

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

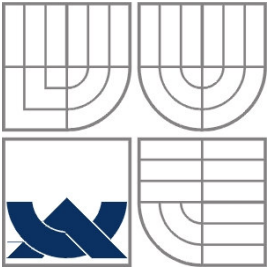
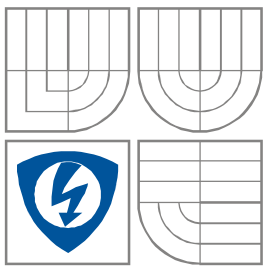
## BEZDRÁTOVÝ ZVONEK S DIGITÁLNÍM PŘENOSEM HLASU

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. PATRIK PARTSCH

BRNO 2008

	<p><b>VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ</b> BRNO UNIVERSITY OF TECHNOLOGY</p>
	<p><b>FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ ÚSTAV RADIOELEKTRONIKY</b></p> <p>FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION DEPARTMENT OF RADIO ELECTRONICS</p>

# BEZDRÁTOVÝ ZVONEK S DIGITÁLNÍM PŘENOSEM HLASU

WIRELESS DOORBELL WITH DIGITAL VOICE TRANSMISSION

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. Patrik Partsch

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Jan Mikulka

BRNO, 2008



# LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

## 1. Pan

Jméno a příjmení: Bc. Patrik Partsch  
Bytem: Velký Týnec Čechovice 81, Olomouc, 783 72  
Narozen (datum a místo): 21. dubna 1983 v Olomouci

(dále jen „autor“)

a

## 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií  
se sídlem Údolní 53, Brno, 602 00  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:  
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika  
(dále jen „nabyvatel“)

### Čl. 1

#### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
  - ☒ diplomová práce
  - ☐ bakalářská práce
  - ☐ jiná práce, jejíž druh je specifikován jako .....
- (dále jen VŠKP nebo dílo)

Název VŠKP: Bezdrátový zvonek s digitálním přenosem hlasu

Vedoucí/ školitel VŠKP: Ing. Jan Mikulka

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: \_\_\_\_\_

VŠKP odevzdal autor nabyvateli\*:

- |  |   |                    |
|--|---|--------------------|
| <input checked="" type="checkbox"/> v tištěné formě      | – | počet exemplářů: 2 |
| <input checked="" type="checkbox"/> v elektronické formě | – | počet exemplářů: 2 |

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

---

\* hodící se zaškrtněte

## Článek 2

### Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☒ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3

### Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 30. května 2008

.....  
Nabyvatel

.....  
Autor

## **Abstrakt**

Tato diplomová práce se zabývá návrhem bezdrátového zvonku s digitálním přenosem hlasu. Zahrnuje rozbor obvodu nRF9E5. Ve své úvodní části obecně charakterizují daný problém. Ve své další části popisují návrh schématu, návrh desky plošného spoje a návrh software.

nRF9E5 je jednoduchý čip s plně zabudovaným RF transceiverem, 8051 kompatibilním mikrokontrolerem a čtyřmi desetibitovými analogově-digitálními převodníky.

Přes studie veškeré dostupné literatury jsem našel jen velmi málo podrobných informací týkajících se této problematiky.

Hlavním cílem mé diplomové práce je zhotovit bezdrátový zvonek s digitálním přenosem hlasu. Kromě toho práce obsahuje popsání schémat.

## **Klíčová slova**

Mikroprocesor, digitální přenos, hardwarový návrh, softwarový návrh, vzorkovací frekvence.

## **Abstract**

The master's thesis deals with a design and layout of wireless doorbell with digital voice transmission. It also includes the nRF9E5 circuit analysis. In the beginning of the document there is characterization of the problem in general. Next part describes a layout example for the application schematic, a layout of printed circuit board and a layout of software.

nRF9E5 is a true single chip system with fully integrated RF transceiver, 8051 compatible microcontroller and a four input 10 bit AD converter.

Despite of all studies of available literature I have found only very little particular information related to this matter.

In general, the main aim of this work is to make the wireless doorbell with digital voice transmission. Further, illustrated schematics are enclosed as a attachment of this thesis.

## **Keywords**

Microprocesor, digital wireless, hardware layout, software layout, sampling rate.

## **Bibliografická citace práce**

PARTSCH, P. *Bezdrátový zvonek s digitálním přenosem hlasu: diplomová práce*. Brno: FEKT VUT v Brně, 2008. 54 s., Vedoucí diplomové práce Ing. Jan Mikulka.

## Prohlášení

Prohlašuji, že svou diplomovou práci na téma Bezdrátový zvonek s digitálním přenosem hlasu jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 30. května 2008

.....  
podpis autora



## **Poděkování**

Chtěl bych poděkovat všem, kteří mi pomohli s přípravou této diplomové práce. Zejména mé velké díky patří Ing. Radkovi Václavíkovi za odborné vedení mé diplomové práce, dále pak Ing. Janu Mikulkovi za zprostředkování cenných informací.

V Brně dne 30. května 2008

.....  
podpis autora

# OBSAH

1. Úvod .....	2
2. Nordic nRF9E5 .....	3
2.1 Mikroprocesor .....	3
2.2 RF Transceiver.....	4
3. Návrh bezdrátového zvonku s digitálním přenosem hlasu.....	6
3.1 Funkce bezdrátového zvonku s poloduplexním interkomem.....	6
4. Hardwarový návrh vysílacího a přijímacího modulu.....	7
4.1 Řídicí jednotka nRF9E5 s ovládacími prvky .....	7
4.2 Návrh vstupního filtru pro AD převodník.....	9
4.3 Návrh výstupního filtru s NF zesilovačem .....	12
4.4 Napájení vysílacího (TX) a přijímacího (RX) modulu.....	13
4.5 Návrh desky plošných spojů.....	16
5. Softwarový návrh vysílacího a přijímacího modulu .....	17
5.1 Funkce programů .....	17
5.2 Princip vysílání a příjmu obvodu nRF9E5 .....	21
6. Závěr.....	22
Literatura.....	23
Seznam použitých zkratk.....	24
Příloha .....	25
I. Obsah přiloženého CD .....	25
II. Univerzální schéma vysílacího i přijímacího modulu.....	26
III. Výkresy desek plošných spojů a osazovací plánek.....	27
IV. Programátor EEPROM paměti.....	33
V. Výpis programu přijímacího a vysílacího modulu.....	34

# 1. Úvod

Předkládaná práce se zabývá návrhem bezdrátového zvonku s digitálním přenosem hlasu. Konkrétní bezdrátový zvonek je řízený obvodem nRF9E5 firmy Nordic, který bude komunikovat v bezlicenčním pásmu na kmitočtu 433 Mhz.

Součástí práce je také nastudování obvodu nRF9E5 a zhotovení schématu zapojení vysílacího a přijímacího modulu s tímto obvodem. V předkládané práci také představuji návrh desky plošného spoje k tomuto bezdrátovému zvonku a software určený pro řízení vysílacího a přijímacího modulu.

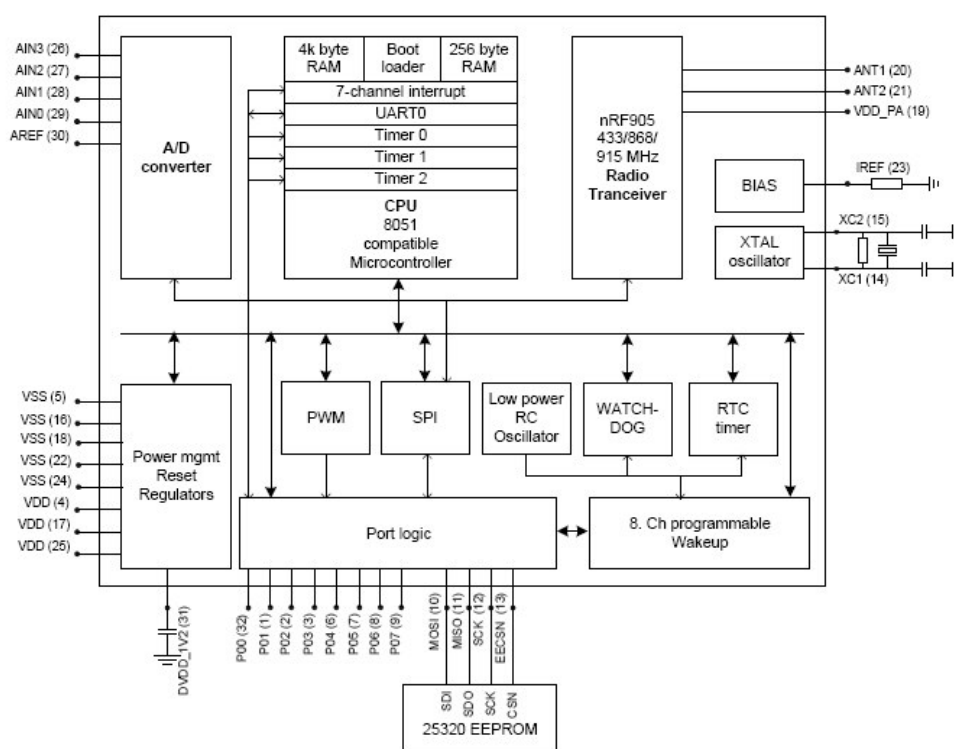
Cílem mé diplomové práce je nejen sestavení bezdrátového zvonku se signalizací zvonění, ale také využití tohoto zvonku jako poloduplexního interkomu. Digitalizace hlasu bude provedena vestavěným AD převodníkem s rozlišením 8 bitů, rekonstrukce audia pak pomocí PWM modulu a aktivního filtru. Aktivní filtr je dolní propust 3. řádu, zapojení Sallen Key, který dostatečně potlačí vzorkovací kmitočet PWM a zaručí nízkou výstupní impedanci. Pro správnou funkci AD převodníku je navržen vstupní filtr dolní propusti s mezní frekvencí 2,4 kHz. Tento filtr je simulován v programu PSpice, kde je provedena toleranční analýza Monte Carlo a Worst Case. Rekonstruovaný audio signál je zesílen pro reprodukci pomocí nízkofrekvenčního zesilovače NCP2890 od firmy ON Semiconductor.

V této práci se také zabývám spotřebou vysílacího modulu, který je napájen bateriemi. Vzhledem k tomu byly vybrány ke konstrukci obvody s nízkou spotřebou a možností softwarového vypnutí. K tomuto účelu byl přizpůsoben i software.

Přes veškeré studie dostupné literatury jsem našel jen velmi málo podrobných informací týkajících se této problematiky. Pro zpracování teoretické části jsem také čerpal z manuálu společnosti Nordic a ON Semiconductor, týkající se výše uvedených obvodů.

## 2. Nordic nRF9E5

Nordic nRF9E5 je čip s vysokofrekvenčním transceiverem, 8051 (resp. 8052) kompatibilním mikroprocesorem a 10-ti bitovým AD převodníkem se čtyřmi vstupy. Čip obsahuje i další periferie a to PWM, SPI, watchdog atd. Obvod se vyrábí v malém 32 vývodovém QFN pouzdře. Napájecí napětí obvodu je 1,9 V až 3,6 V.



Obr. 1. Blokové schéma obvodu nFR9E5.

### 2.1 Mikroprocesor

Základem je 8-mi bitový CISC mikroprocesor, který je rozšířen o 5 dalších zdrojů přerušení. Uvedené zdroje jsou ADC, SPI, dva zdroje přerušení pro řízení transceiveru a jeden pro wakeup režim. Mikroprocesor obsahuje i tři časovače a UART.

Mikroprocesor obsahuje interní 4kB paměť RAM pro software. Program se do této interní RAM paměti nahrává z externí EEPROM paměti přes sběrnici SPI takzvaným BOOTLOADEREM. Ten je uložen v paměti ROM o velikosti 512 bajtů.

Jako externí paměť se musí použít EEPROM typu 25320 s SPI rozhraním. Dále mikroprocesor využívá 256 bytů RAM pro data a speciální funkční registry, které jsou rozšířené pro řízení několika výjimečných funkcí.

Obvod obsahuje pulzně šířkovou modulaci (PWM) s rozlišením 6, 7 nebo 8 bitů. Frekvence PWM je odvozena z krystalu mikroprocesoru přes 6-ti bitový předdělič. Činitel plnění je pak 0-100% a řídí se jedním 8-mi bitovým registrem.

Sběrnice SPI je typu "master" a je multiplexovaná mezi výstupními piny, AD převodníkem a RF transceiverem.

V obvodu nRF9E5 je k dispozici 5-ti kanálový 10-ti bitový AD převodník umožňující maximální vzorkovací kmitočet 80 ksps (80000 vzorků za sekundu) pro jeden vybraný kanál. První čtyři kanály jsou dostupné na pouzdře obvodu a poslední kanál je interní, který měří napájecí napětí obvodu. Referenční napětí může být použito interní 1.22 V nebo externí od 0 V do 1,5 V. Rychlost převodníku je řízena programem stejně tak jako rozlišení, které může být nastaveno na 6, 8, 10 či 12 bitů.

Výhodou obvodu je, že může být naprogramován do několika stavů s různou proudovou spotřebou. Jednotlivé subsystémy jako ADC či RF transceiver mohou být programově vypnuty či zapnuty. Mikroprocesor se zastaví, ale hodnoty paměti a registrů zůstávají beze změny. Proudová spotřeba je typicky 2,5 uA a režim je možné ukončit přerušením z watchdogu, RTC časovače či portu P0.

nRF9E5 má dva porty P0 a P1. Porty jsou obousměrné, na portu P0 je dostupných 8 pinů a na P1 pak další 4 piny sběrnice SPI. Ke každému pinu může být také přiřazena alternativní funkce.

## 2.2 RF Transceiver

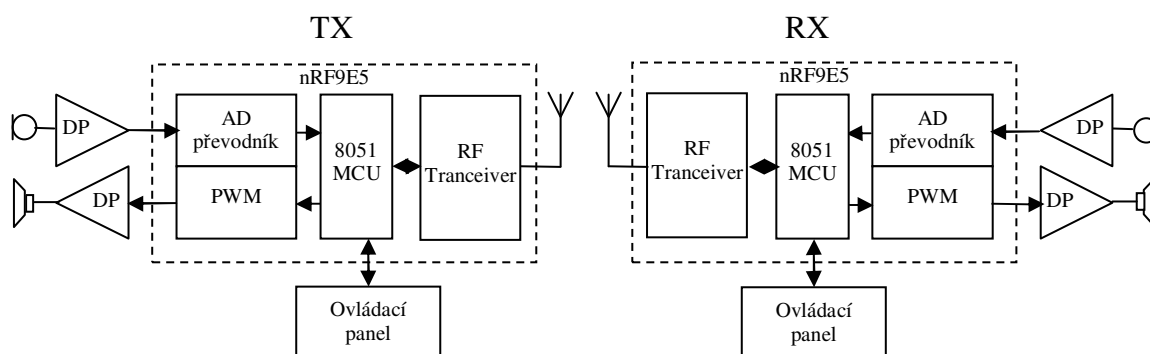
Obvod vysokofrekvenčního transceiveru je funkčně shodný s obvodem nRF905. S mikroprocesorem komunikuje po interní paralelní nebo sériové SPI sběrnici. Obvod je určen pro práci v ISM pásmech 433/868/915MHz s různým počtem kanálů. Transceiver se skládá z integrovaného frekvenčního syntezaátoru, výkonového zesilovače, modulátoru a přijímače. Výstupní výkon, kmitočet a další vysokofrekvenční parametry se dají snadno modifikovat přes SPI sběrnici. Maximální vysílací výkon může být nastaven až do hodnoty 10dBm. S čtvrtvlnným dipólem je dosah obvodu cca 300 m ve volném prostředí.

Transceiver užívá GFSK modulaci a Manchester kódování se stálou přenosovou rychlostí 50 kbps. Pro snížení spotřeby při vysílání se užívá režimu ShockBurst (TM). Je to efektivní využití maximální dostupné komunikační rychlosti obvodu nRF9E5. Data s nižší přenosovou rychlostí jsou přenášena po sběrnici SPI do obvodu RF transceiveru, který je po naplnění vyrovnávací paměti vysílá rádiovým kanálem rychlostí až 100 kbps. Díky tomu se redukuje spotřeba proudu během vysílání. Další důležitou vlastností transceiveru je automatické generování hlavičky a CRC (kontrolní součet) pro každý rámeček. To snižuje požadavky na vlastní mikroprocesor i na vývoj software. Uživatel si jen "vyszvedne" data z registrů.

Proudová spotřeba obvodu při vysílání je 11 mA s výstupním výkonem -10 dBm a při příjmu je spotřeba 12,5 mA. RF Transceiver se dá programově vypnout a snížit tak celkovou spotřebu zařízení. Výstupní výkon se také dá programově řídit ve čtyřech stupních -10 dBm (11 mA), -2 dBm (14 mA), 6 dBm (20 mA) a 10 dBm (30 mA).

### 3. Návrh bezdrátového zvonku s digitálním přenosem hlasu

Celá realizace bezdrátového zvonku s funkcí poloduplexního interkomu je na bázi mikroprocesoru s vysokofrekvenčním čipem nRF9E5 firmy Nordic. Na Obr. 2. je blokové schéma bezdrátového zvonku s digitálním přenosem hlasu. Bezdrátový zvonek se dělí na dvě hlavní části, na vysílací modul (TX) a přijímací modul (RX). Vysílací modul bude umístěn vně budovy a přijímací modul uvnitř budovy. Digitalizace hlasu bude prováděna vestavěným AD převodníkem s rozlišením 8 bitů a rekonstrukce audia pak pomocí PWM modulu a aktivního filtru.



Obr. 2. Blokové schéma bezdrátového zvonku s digitálním přenosem hlasu.

#### 3.1 Funkce bezdrátového zvonku s poloduplexním interkomem

Vysílací (TX) modul je vybaven mikrofonom a reproduktorem pro přenos a příjem řečového signálu. Obsahuje také ovládací panel s jedním spínačem pro aktivaci vyzváněcího tónu na přijímací straně (RX). Přijímací modul (RX) je také vybaven mikrofonom, reproduktorem a ovládacím panelem. Ovládací panel obsahuje dva spínače a jednu LED diodu. První spínač je použit pro přenos řečového signálu, druhý pro otvírání dveří. Dioda LED na přijímači slouží pro signalizaci potřeby výměny baterií na vysílací straně.

Vysílací modul je napájen bateriemi, proto je zapotřebí zajistit co nejnižší spotřebu energie. Toho se docílí tím, že je vysílací modul v režimu spánku. V tomto modu spánku zůstává do té doby než přijde účastník k vysílacímu modulu a stiskne spínač. Modul vysílače se probudí z režimu spánku, vyšle signál do přijímacího modulu, kde se aktivuje vyzváněcí tón. Poté druhý účastník, na přijímacím modulu, stiskne a drží spínač pro aktivaci poloduplexního interkomu. Poloduplexní interkom zůstává aktivní ve směru přijímač-vysílač po celou dobu sepnutého spínače na přijímací straně. Po uvolnění spínače bude možná automatická komunikace ve směru vysílač-přijímač. Poté účastník na přijímací straně stiskne druhé tlačítko pro otevření dveří, a tím i uspí vysílací modul.

## 4. Hardwarový návrh vysílacího a přijímacího modulu

Přijímací i vysílací modul je rozdělen na tři části. První částí je řídicí jednotka nRF9E5 s ovládacími prvky, druhá je vstupní filtr pro AD převodník a poslední částí je výstupní filtr s NF zesilovačem. Tyto tři části jsou u obou dvou modulů skoro totožné. Výjimkou je první část, kde přijímač obsahuje navíc jeden spínač a LED diodu a vysílací část obsahuje navíc relé pro otevírání vstupních dveří.

### 4.1 Řídicí jednotka nRF9E5 s ovládacími prvky

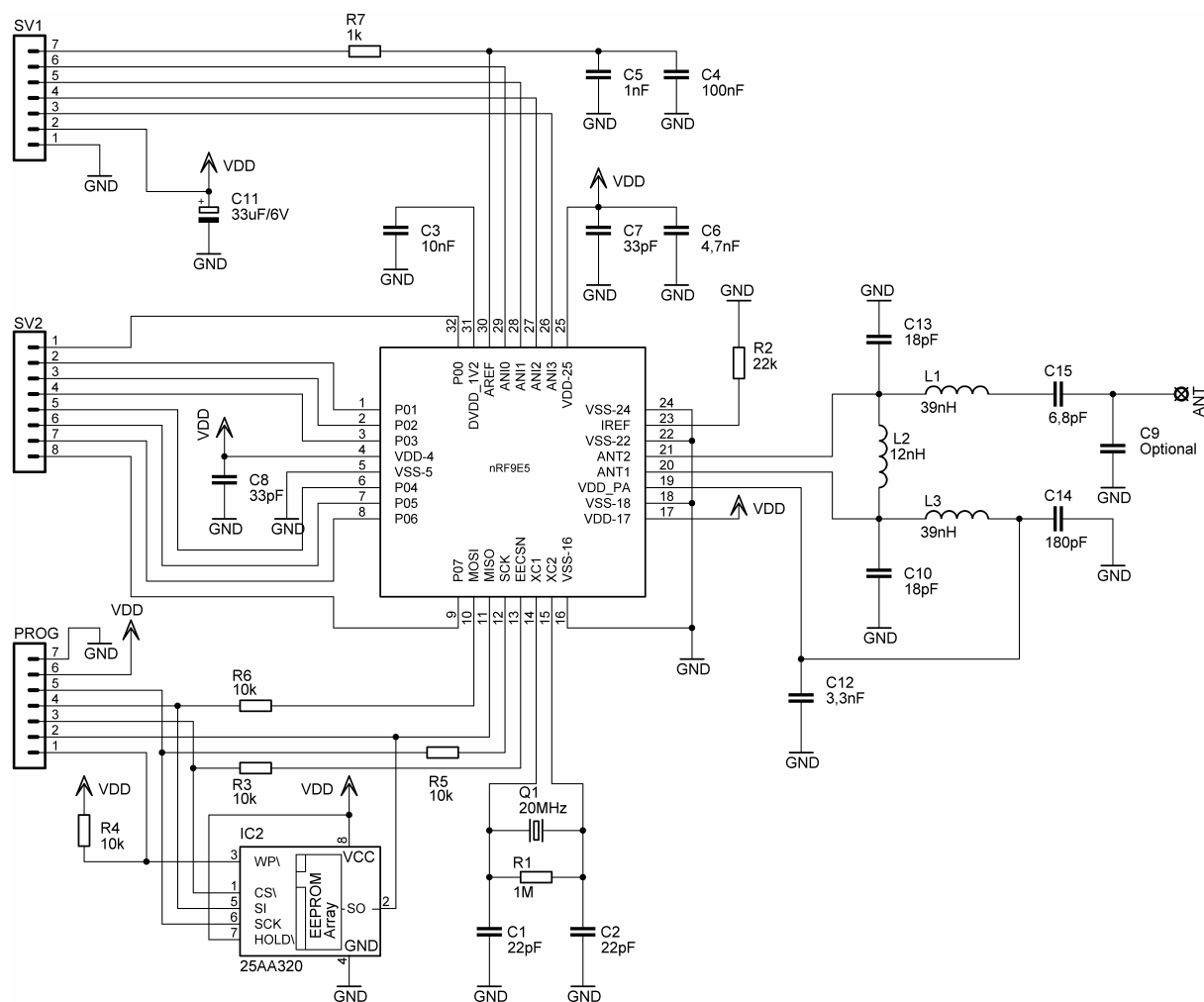
Hlavní částí řídicí jednotky je mikroprocesor s vysokofrekvenčním čipem nRF9E5, který se stará o bezdrátovou komunikaci a převod analogového hlasu na digitální a zpět. Stará se též o vyhodnocování stisknutých spínačů a stavu napájecího napětí na vysílacím modulu.

K převodu analogového hlasu na digitální je zapotřebí 8-mi bitový AD převodník (viz. kapitola 4.2 Návrh vstupního filtru pro AD převodník). Zdroj referenčního napětí převodníku je použit interní, z obvodu nRF9E5 o hodnotě 1,22 V. K vyhodnocování stavu napájecího napětí na vysílacím modulu je použit interní převodník VDD/3, který vyžaduje interní referenční zdroj. Protože napájecí napětí obvodu nRF9E5 může být maximálně 3,6 V je před převodníkem zabudován interní dělič třemi. Tudíž při maximálním napájecím napětí 3,6 V, je na VDD/3 převodníku 1,2 V. Při nízkém napájecím napětí cca 2,1 V vyše vysílací modul signál do přijímacího modulu a ten rozsvítí LED1. Z tohoto důvodu je přijímací modul navíc vybaven svítící diodou LED1.

Pro převod z digitálního hlasu na analogový je použit PWM modul s rozlišením 8 bitů, aktivní filtr a nízkofrekvenční zesilovač (viz. kapitola 4.3 Návrh výstupního filtru s NF zesilovačem).

Schéma řídicí jednotky nRF9E5 s ovládacími prvky je na Obr. 3. Schéma představuje katalogové zapojení obvodu pro práci v pásmu 433 MHz. Tlumivky L1 až L3 spolu s kondenzátory C10, C12, C13, C14 a C15 tvoří transformační obvod pro připojení 50  $\Omega$  antény. Krystal Q1, rezistor R1 a kondenzátory C1 a C2 tvoří obvod oscilátoru. Kondenzátor C3 až C7 tvoří blokovací kondenzátory, R2 je přesný jednaprocentní referenční odpor použitý pro nastavení interních obvodů nRF9E5. Rezistor R7 spolu s kondenzátory C5 a C4 slouží k připojení externí reference pro analogově digitální převodník. IC2 je sériová paměť EEPROM 25AA320. Rezistory R6 a R5 připojené k paměti 25AA320 slouží jako ochrana nRF9E5 při programování paměti. Paměť 25AA320 se programuje pomocí SPI signálů vyvedených na konektor PROG. Tato paměť 25AA320 se dá programovat pomocí programátoru PonyProg s jednoduchým obvodem, který je přiložen v příloze IV. Programátor EEPROM paměti.





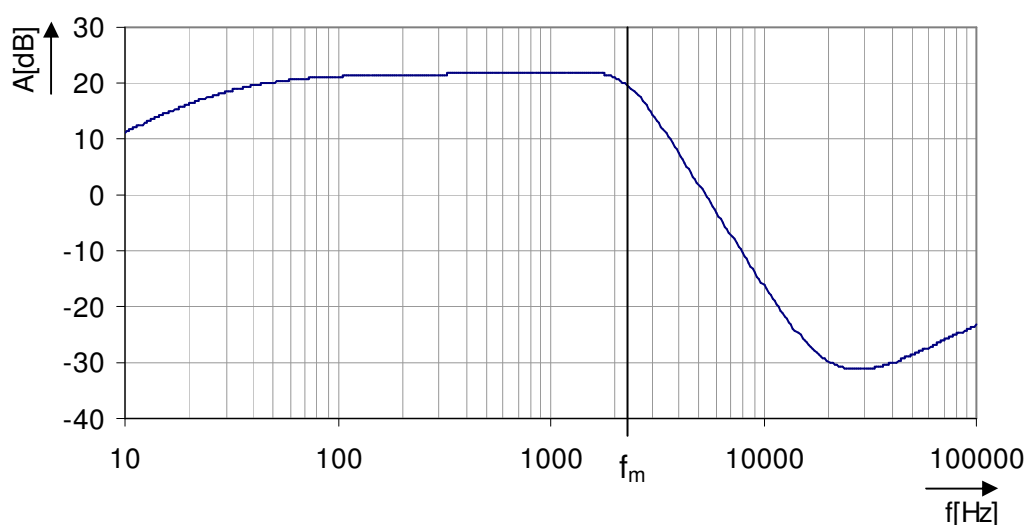
Obr. 3. Schéma řídicí jednotky nRF9E5

Jak již bylo popsáno výše, vysílací modul obsahuje jeden spínač a relé pro otevírání vstupních dveří. Příjemací modul obsahuje dva spínače a jednu LED diodu. Tyto ovládací prvky jsou zobrazeny ve schématu na Obr. II.1 v příloze II. Univerzální schéma vysílacího i přijímacího modulu. Protože jak vysílací tak přijímací moduly jsou z velké části totožné je vytvořeno jedno univerzální schéma pro oba moduly (viz Obr. II.1). Spínače S1 a S2, připojené na port P02 a P03, obsahují zvedací (pull-up) rezistory R19 a R24. Zákmity způsobené spínači jsou odstraněny integračním článkem z R20 (R25) a C16 (C18). LED1 dioda je přímo buzena z portu P01 obvodu nRF9E5 přes rezistor R26 směrem k napájecímu napětí. Přes port P01 u vysílacího modulu je v sérii s rezistorem R22 ovládán tranzistor T1, který spíná relé. K cívce relé je paralelně připojena dioda D1, která eliminuje zákmity.

## 4.2 Návrh vstupního filtru pro AD převodník

Obvod nRF9E5 má bezdrátovou užitečnou komunikační rychlost 39 kbps (viz. kapitola 5. Softwarový návrh vysílacího a přijímacího modulu). Proto je zapotřebí zvukový signál přijímaný z mikrofonu filtrovat dolní propustí s  $f_{mez}=2,4$  kHz. Po filtraci bude analogový signál převeden na digitální pomocí vestavěného AD převodníku s rozlišením 8 bitů. Vzorkovací frekvence AD převodníku musí být alespoň dvojnásobkem nejvyšší frekvence vstupního signálu tedy  $\min f_{vz}=4,8$  kHz. Kdyby byla vzorkovací frekvence menší, docházelo by ke zkreslení složek vyšších frekvencí. Na výstupu AD převodníku je tedy přenosová rychlost 38,4 kbps, která splňuje užitečnou přenosovou rychlost obvodu nRF9E5. Poté se bude tento digitální signál pomocí RF transceiveru v obvodu nRF9E5 vysílat.

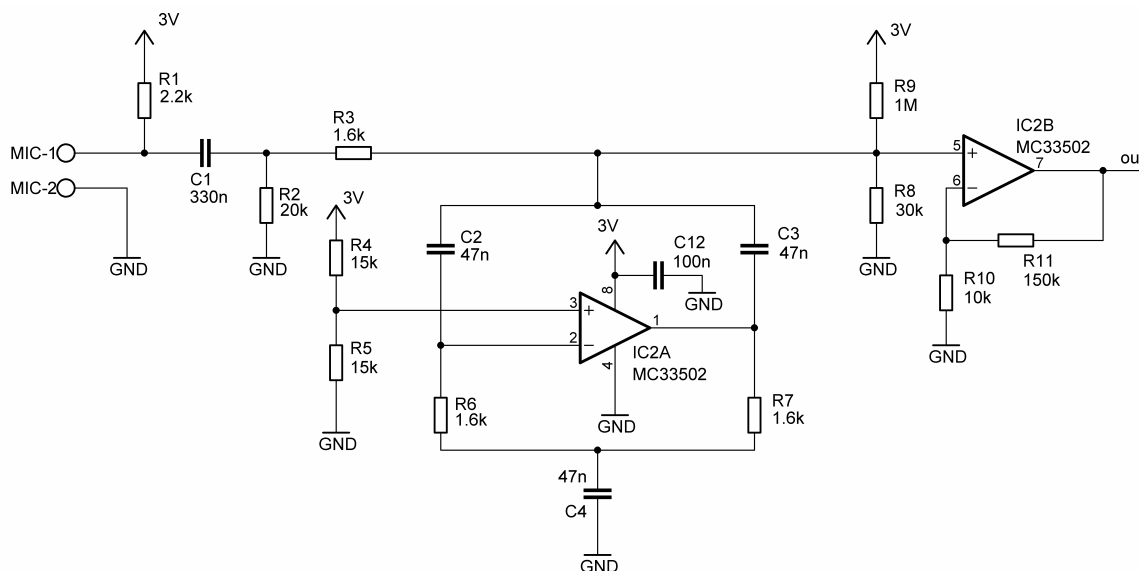
Jako vstupní filtr je použit aktivní filtr RC. Filtr se chová jako dolní propust 3. řádu s mezním kmitočtem  $f_m = 2,4$  kHz. Na Obr. 4 je zobrazena přenosová charakteristika vstupního filtru vytvořená v simulátoru PSpice. Aktivní filtr je navržen pro nesymetrické napájení z důvodu použití bateriového napájení. Použitý je operační zesilovač MC33502 typu RAIL-TO-RAIL. Tento operační zesilovač je vhodný pro nesymetrické napájení a pracuje už od 1 V. Má minimální spotřebu a používá celý rozsah napájecího napětí jak pro vstupní tak i výstupní signály.



Obr 4. Přenosová charakteristika vstupního filtru pro AD převodník

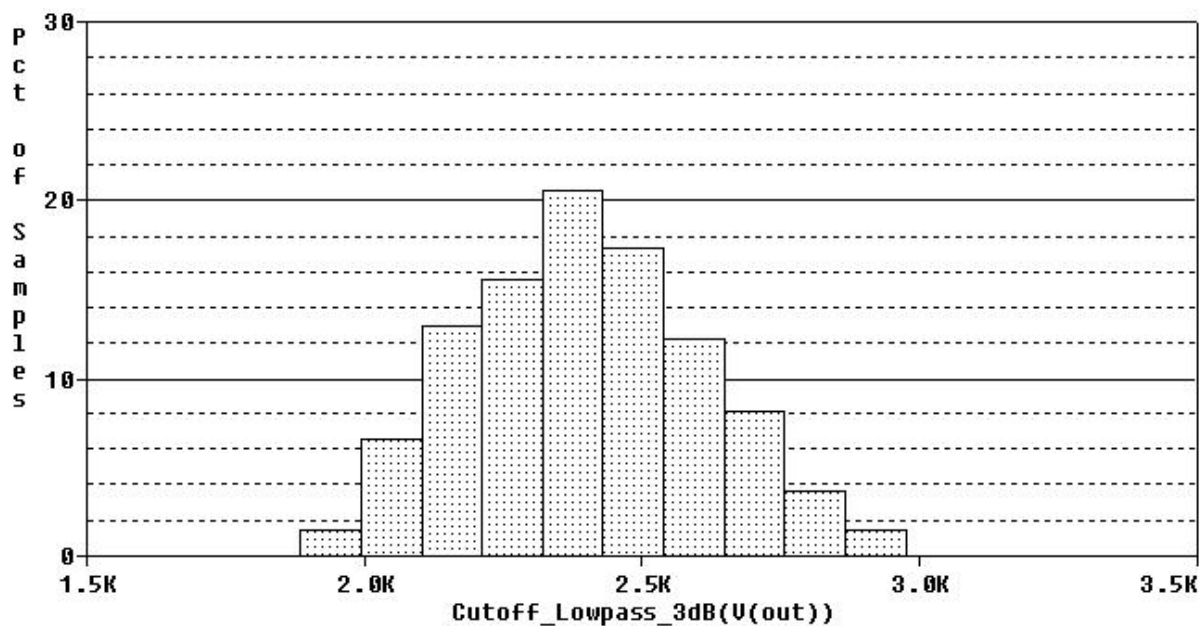
Na vstupu filtru pro AD převodník je elektretový mikrofón napájen přes rezistor R1. Spolu vytváří vstupní signál s maximální amplitudou 75 mV. Kondenzátorem C1 je oddělena nežádoucí stejnosměrná složka která napájí elektretový mikrofón. Protože na vstup AD převodníku je možné přivést maximální kladnou amplitudu 1,2 V, je zapotřebí signál z mikrofónu zesílit. Zesílení se provede neinventujícím

zapojením operačního zesilovače IC2B s rezistory R10 a R11. Přidáním stejnosměrné složky rezistory R9 a R8 se vytvoří střední hodnota 0,6 V pro AD převodník. Operační zesilovač IC2A a pasivním prvky vytváří dolní propust 3. řádu. Výhodou tohoto zapojení je, že vstupy a výstupy operační zesilovač jsou galvanicky izolovány vazebními kapacitami C2 a C3 od hlavní cesty signálu. Rezistory R4 a R5 tvoří umělý střed napájení pro operační zesilovač.



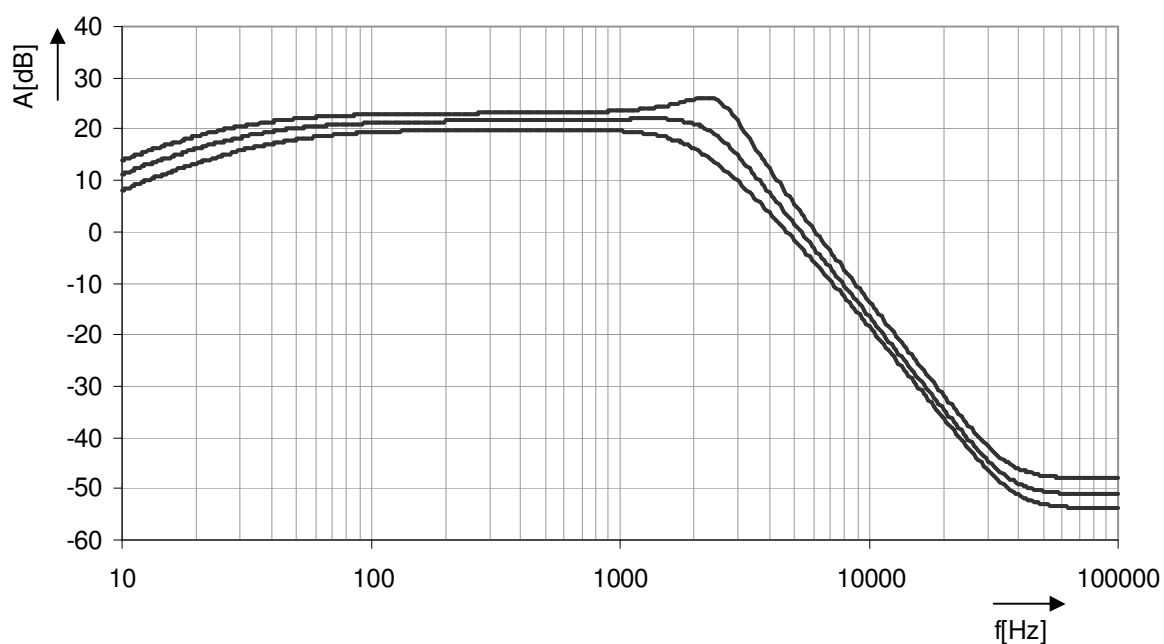
Obr. 5. Schéma vstupního filtru pro AD převodník

Vstupní filtr je navržen s SMD součástkami, které mají velký rozptyl kolem nominální hodnoty. U rezistorů to dělá  $\pm 5\%$  a u kondenzátorů až  $\pm 20\%$ . Pro zjištění jaký vliv má tato tolerance hodnot na přenosovou charakteristiku filtru, byla provedena toleranční analýza Monte Carlo a Worst Case programem PSpice. Během analýzy Monte Carlo jsou parametry vybírány náhodně z tolerančního intervalu hodnot. Pro každý výběr se stanoví hledaná funkce obvodu tj. mezní kmitočet. Analýza byla provedena pro 1000 výběrů. Výsledkem je histogram (Obr. 6 Histogram analýzy Monte Carlo), který zobrazuje odhad v jakém intervalu bude ležet s největší pravděpodobností mezní kmitočet. Kde výška každého sloupce udává četnost v procentech, že sledovaná veličina leží v intervalu určeném šířkou sloupce.



Obr. 6. Histogram analýzy Monte Carlo

V další toleranční analýze Worst Case byl zjištěn nejhorší možný rozptyl od nominální hodnoty mezního kmitočtu  $f_{mez}=2,4$  kHz. Analýza Woert Case v podstatě hledá takové rozložení hodnot parametrů součástek v rámci tolerančních intervalů, které znamená nejhorší možnou hodnotu sledované veličiny. Výsledek analýzy je na Obr. 7, kde prostřední přenosová charakteristika je nominální a postraní postranní přenosové charakteristiky jsou nejhorší možné.

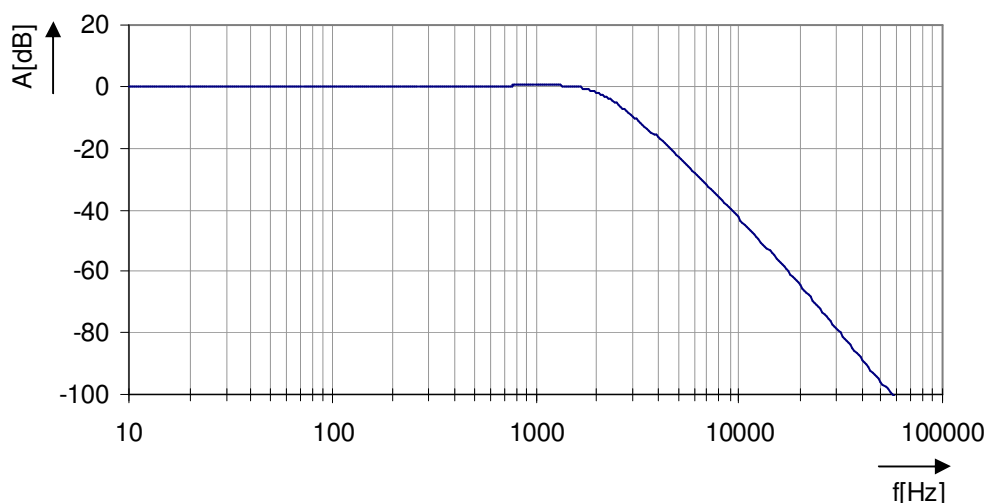


Obr. 7. Přenosová charakteristika vstupního filtru s nejhorším rozptylem od nominální hodnoty

Z provedených tolerančních analýz vyplývá, že nejhorší nožné hodnoty mezního kmitočtu mohou být  $f_{mez} = 1,86 \text{ kHz}$  a  $f_{mez} = 2,80 \text{ kHz}$ , ale s malou pravděpodobností.

### 4.3 Návrh výstupního filtru s NF zesilovačem

Přijímaný digitální zvukový signál je potřeba převést zpět do analogové podoby a dostatečně zesílit pro vybuzení malého reproduktoru. Protože obvod nRF9E5 nemá DA převodník, použil jsem PWM modul s rozlišením 8 bitů a aktivním filtrem RC. Aktivní filtr je dolní propust 3. řádu, zapojení Sallen Key, který dostatečně potlačí vzorkovací kmitočet PWM a zaručí nízkou výstupní impedanci. Přenosová charakteristika výstupního filtru je na Obr. 8

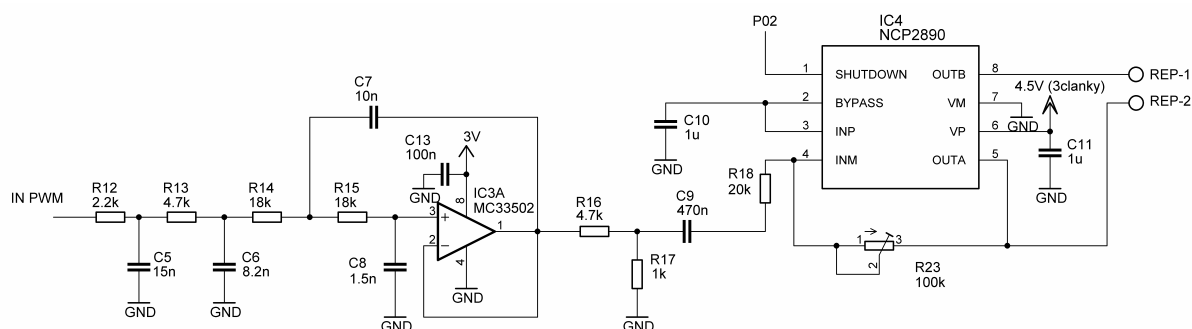


Obr 8. Přenosová charakteristika výstupního filtru

Jako nízkofrekvenční zesilovač je použit obvod NCP2890. Napájecí napětí obvodu NCP2890 je v rozmezí od 2,2 V do 5,5 V. Při tomto napájecím napětí má výstupní výkon cca od 300 mW do 1 W s  $8 \Omega$  reproduktorem. Zapojení nízkofrekvenčního zesilovače přímo vychází z katalogového zapojení výrobce. Na vstup zesilovače je sériově zapojen kondenzátor C9, který odstraní stejnosměrnou složku. Rezistor R18 a trimr R23 vytváří odporový dělič se kterým dá regulovat hlasitost. Obvod NCP2890 obsahuje i pin shutdown, přivedený na port P02 obvodu nRF9E5, kterým se dá zesilovač softwarově vypnout. Při tomto vypínacím módu má zesilovač minimální odběr 10 nA (viz. kapitola 4.4 Napájení vysílacího (TX) a přijímacího (RX) modulu).

Mezi aktivním filtrem se zesílením 0 dB a nízkofrekvenčním zesilovačem je napěťový dělič R16/R17 s útlumem 15 dB. Tento dělič napětí je zde zařazen kvůli výstupnímu napětí na PWM modulu. PWM modul má amplitudu od 0 V do hodnoty napájecího napětí obvodu nRF9E5, tj. při bateriovém napájení cca 3 V. Tudíž PWM

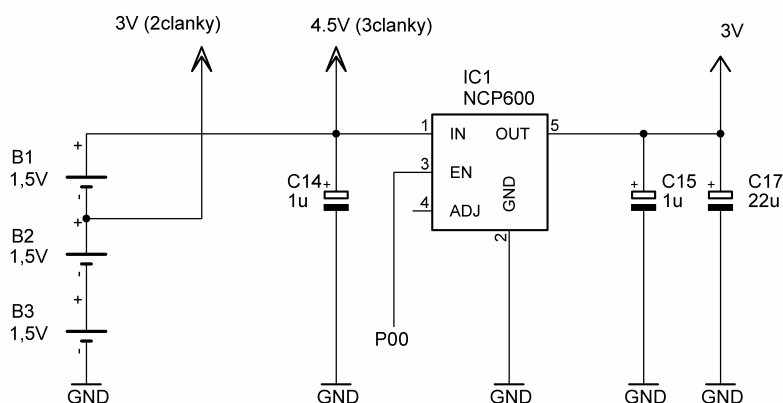
signál projde aktivním filtrem na vstup napěťového děliče. Zde se sníží na hodnotu cca 0,53 V a přivede do nízkofrekvenčního zesilovače, který požaduje maximální vstupní napětí 0,6 V<sub>PP</sub>.



Obr. 9. Schéma výstupního filtru s NF zesilovačem

#### 4.4 Napájení vysílacího (TX) a přijímacího (RX) modulu

Přijímací modul bude umístěn v budově, kde bude napájen ze zdroje napětí připojeného na rozvodnou elektrickou síť. Oproti tomu vysílací modul bude umístěn mimo budovu, proto je zapotřebí napájení z tužkových baterií. Jak vysílací tak přijímací modul potřebuje tři různé zdroje napětí pro napájení různých částí v obvodu. Řídící jednotka nRF9E5 pracuje v rozmezí napětí 1,9 V až 3,6 V. Nízkofrekvenční zesilovač potřebuje napětí v rozmezí 2,2 V až 5,5 V, a vstupní a výstupní filtry potřebují přesnou hodnotu 3V. Tedy pro napájení vysílacího modulu bude zapotřebí tři článků (tužkových baterií) zapojených do série. Třemi články bude napájen nízkofrekvenční zesilovač NCP2890 a stabilizátor napětí NCP600, který vytváří konstantní hodnotu 3 V. Dvěma články bude napájena řídicí jednotka nRF9E5. Rozvod napájecího napětí ze tří článků je zobrazen ve schématu na Obr. 10. Schéma napájení modulů.



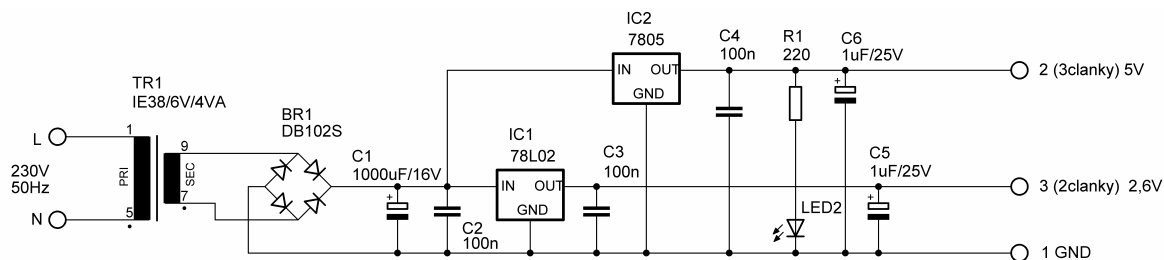
Obr. 10. Schéma napájení modulů

Řídicí jednotka nRF9E5 u vysílacího modulu je přímo napájena bateriemi z důvodu zjištění stavu baterií. Obvod nRF9E5 hlídá hodnotu napájecího napětí pomocí interního  $VDD/3$  převodníku. Pokud se baterie vybije pod hodnotu 1,1 V na článek, vysílací modul vyšle signál přijímacímu modulu, ten bude signalizovat rozsvícením diody LED1, že je nutná výměna baterií ve vysílacím modulu.

Jak již bylo napsáno stabilizátor NCP600 stabilizuje bateriové napětí na hodnotu 3 V a napájí vstupní filtr, výstupní filtr a mikrofon. Stabilizátor NCP600 v modulu je použit ze dvou důvodů. První je, že mikrofon je potřeba napájet konstantní hodnotu, jinak by se zmenšováním napětí zmenšovala amplituda výstupního signálu z mikrofonu. Druhý důvod je, že stabilizátor NCP600 má zabudován vypínací mód, který je přiveden na pin P00 obvodu nRF9E5. Tento vypínací mód slouží ke snížení odběru z baterií při jeho nečinnosti.

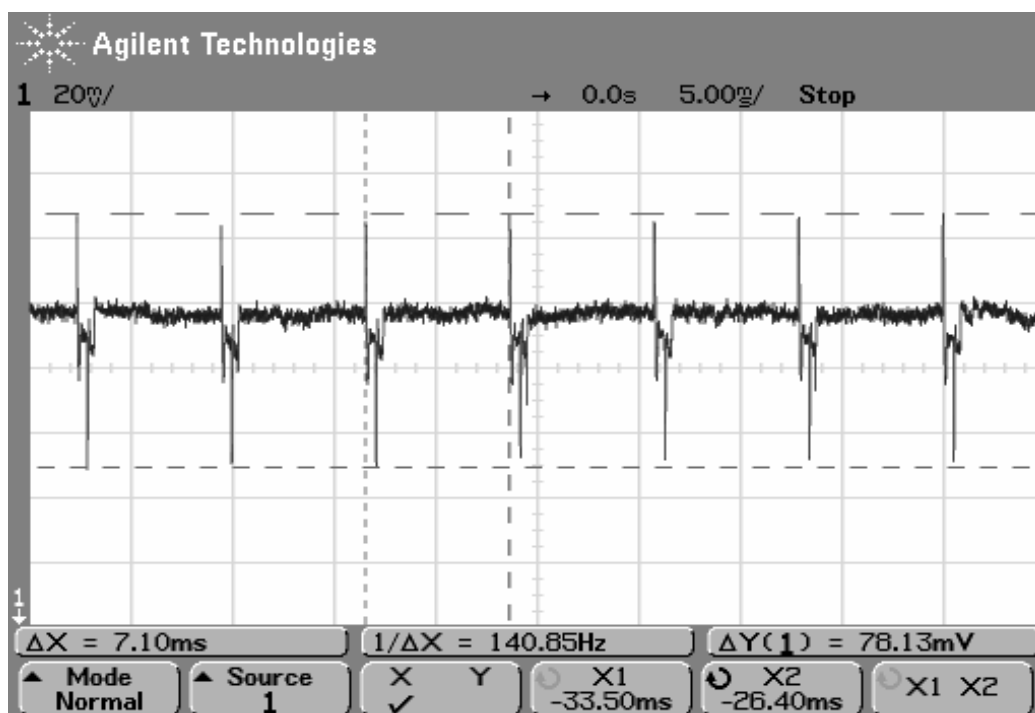
Protože vysílací modul je napájen z baterií je potřeba brát ohled na nízkou spotřebu modulu. Toho se docílí vypínáním různých částí modulu při nečinnosti. Nejmenší odběr bude mít modul před stiskem spínače (zvonění). Řídicí jednotka nRF9E5 bude v režimu Power Down Mode s odběrem 2,5 uA, nízkofrekvenční zesilovač NCP2890 bude mít při vypnutém stavu odběr jen 10 nA a u stabilizátor NCP600 je to při vypnutém stavu také 10 nA. Po stisknutí spínače se probudí řídicí jednotka nRF9E5 z režimu Power Down Mode a začne vysílat signál pro zvonění na přijímací modul. Při tom se zvýší její odběr až na 30 mA. Poté nastane komunikace mezi účastníky, kde řídicí jednotka nRF9E5 aktivuje stabilizátor NCP600 a nízkofrekvenční zesilovač NCP2890. Nízkofrekvenční zesilovač se bude během komunikace vypínat a zapínat v závislosti na směru komunikace. Při tom může být odběr až 500 mA v závislosti na hlasitosti reproduktoru. Po ukončení komunikace, stisknutí spínače S2 na přijímacím modulu, se vysílací modul vrátí do úsporného režimu.

Přijímací modul též využívá vypínání a zapínání stabilizátoru NCP600 a nízkofrekvenčního zesilovače NCP2890, ale nevyužívá režimu Power Down Mode u řídicí jednotky nRF9E5. Důvod proč se nevyužívá režim Power Down Mode je ten, že přijímací modul musí být stále na příjmu a čekat než začne vysílací modul vysílat. Při tomto čekání má řídicí jednotka nRF9E5 odběr 12,5 mA. Z tohoto důvodu je nevhodné přijímací modul napájet bateriemi. Schéma síťového zdroje pro napájení přijímacího modulu je zobrazeno na Obr. 11. Schéma síťového zdroje. Síťové napětí 230 V je přivedeno na transformátor TR1. Transformátor má jedno sekundární vinutí 6 V a je schopen dodávat proud až do 660 mA. Střídavé napětí je dále přivedeno do usměrňovacího můstku BR1. Následuje elektrolytický kondenzátor C1, který filtruje procházející napětí. Filtrované napětí je dále rozděleno do dvou větví +5 V pro napájení nízkofrekvenčního obvodu NCP2890 a stabilizátoru NCP600 a +2,6 V pro napájení řídicí jednotky nRF9E5. Tyto napětí jsou získána pomocí stabilizátoru řady 78xx (IC1, IC2) v klasickém doporučeném zapojení. Protože ze stabilizátoru IC2 (7805) bude při nečinnosti nízkofrekvenčního zesilovače NCP2890 odebírán malý proud je v této větvi zapojen zatěžovací rezistor R1. Tento zatěžovací rezistor zajistí minimální odběr 20 mA z obvodu pro správnou funkci stabilizace. V sérii s rezistorem R1 je zapojena led dioda LED2, která signalizuje přítomnost napětí.



Obr. 11. Schéma síťového zdroje

Při návrhu a testování bezdrátového zvonku s digitálním přenosem hlasu se při přenosu hlasu objevovalo velmi silné nežádoucí rušení (viz Obr. 12. Záznam o rušení při přenosu hlasu pomocí osciloskopu). Rušení bylo naměřeno za výstupem stabilizátoru NCP600 a ovlivňovalo tak vstupní filtr. Měřením bylo zjištěno, že toto rušení má pravidelnou periodu cca 140 Hz tj. každých 7 ms se objevoval rušivý impuls. Po čase bylo zjištěno, že rušení způsobuje obvod nRF9E5 svým vysíláním. Protože obvod nRF9E5 nevysílá spojitě, ale po určitém časovém úseku tj. vysílač se zapíná a vypíná. Tím vznikly na napájecích přívodech tyto rušivé impulsy. Rušení bylo odstraněno zapojením filtrujícího kondenzátoru C17 (viz. Obr. 10. Schéma napájení modulů) co nejblíže k napájení filtrů. K filtrování byl použit kvalitní kondenzátor s malou hodnotou ESR (ekvivalentní sériový odpor) 50 mΩ. Tím rušení způsobené vysíláním vymizelo.



Obr. 12. Záznam o rušení při přenosu hlasu pomocí osciloskopu



## 4.5 Návrh desky plošných spojů

Přijímací i vysílací modul je navržen na dvou univerzálních oboustranných plošných spojích. První univerzální plošný spoj obsahuje vstupní filtr, výstupní filtr, nízkofrekvenční zesilovač, napájení a ovládací prvky. Plošný spoj je univerzální v tom, že se dá použít jak na přijímací část tak na vysílací část. Rozdíl bude jen v osazení součástek (viz. Příloha III. Výkresy desek plošných spojů a osazovací plánek). Druhý univerzální plošný spoj obsahuje řídicí jednotku nRF9E5. Tento plošný spoj je použit na přijímací i vysílací části. Plošný spoj řídicí jednotky nRF9E5 byl převzat [3]. Oba plošné spoje budou propojeny propojovacími lištami (SV1, SV2 a SV3). Plošné spoje budou pracovat s vysokofrekvenčními obvody, tomu odpovídá oboustranná zemnicí plocha propojená vhodně rozmístěnými via průchody. Všechny součástky jsou v provedení SMD, pasivní součástky jsou v rozměrech 0805.

Plošný spoj síťového zdroje pro přijímací modul je jednostranný a používá z velké části klasické velikosti součástek. Síťový zdroj je k přijímacímu modulu připojen třemi přívodními kabely. Plošný spoj síťového zdroje je v příloze III. Výkresy desek plošných spojů a osazovací plánek.

## 5. Softwarový návrh vysílacího a přijímacího modulu

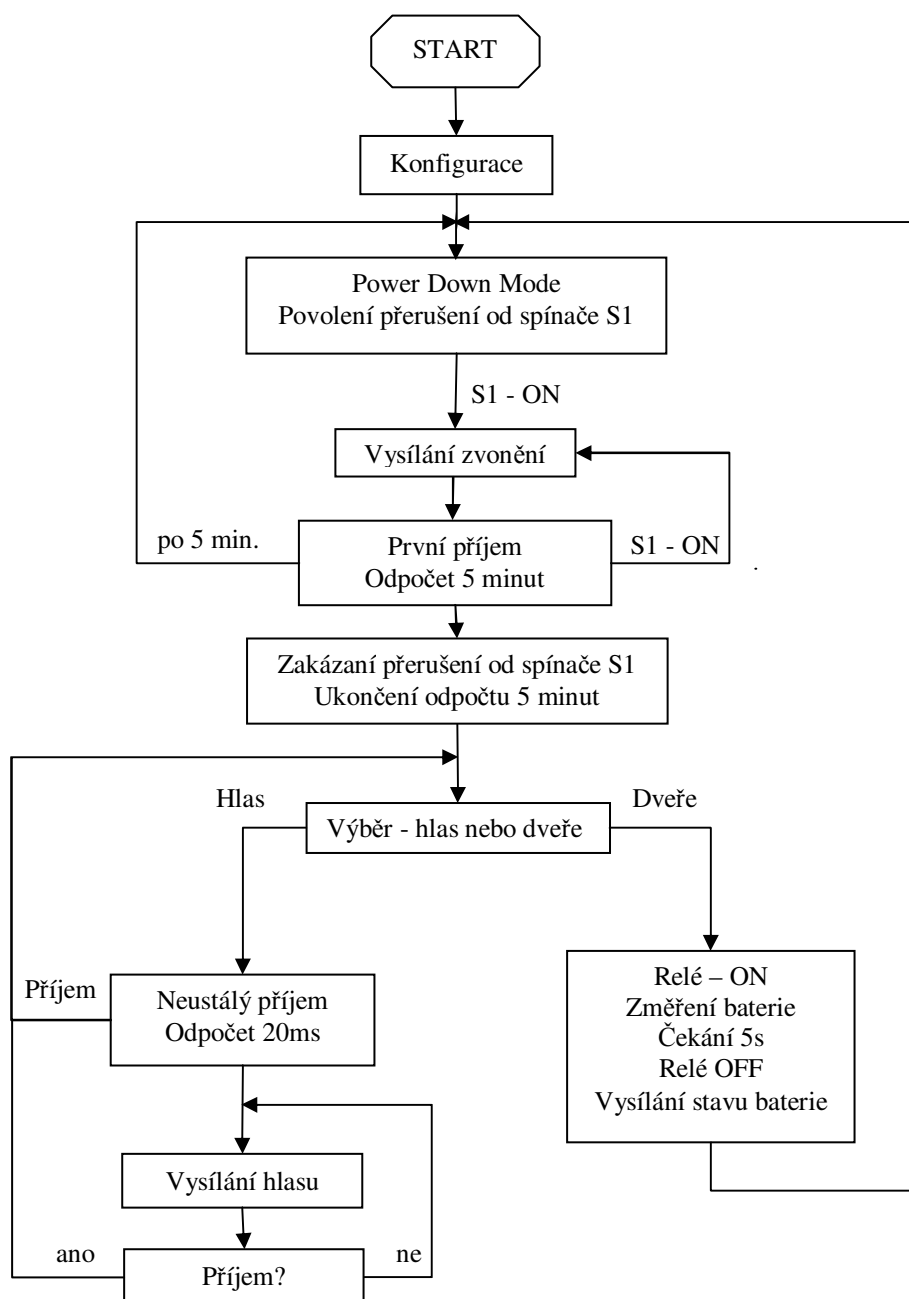
Řídící jednotka nRF9E5, přijímacího i vysílacího modulu, obsahuje software, který řídí hardware bezdrátového zvonku s digitálním přenosem hlasu. Přijímací i vysílací modul mají svůj vlastní rozdílný software. Navržený program pro vysílací modul obsluhuje bezdrátovou komunikaci, spínač S1, úsporný režim vysílacího modulu, spínání relé, měření stavu napájecí baterie a jiné řídicí funkce pro správnou činnost modulu. Program pro přijímací modul obsluhuje také bezdrátovou komunikaci, spínače S1 a S2, LED diodu pro zobrazení stavu napětí na vysílacím modulu a jiné. Programy byly programovány v jazyce C. Zdrojové texty programů s popisem jsou přiloženy v příloze V. Výpis programu přijímacího a vysílacího modulu.

### 5.1 Funkce programů

Oba programy začínají konfigurací. Zde se nastavuje PWM modul, AD převodník, SPI sběrnice, časovače T1 a T2, bezdrátová komunikace a nastavení portů P0. Program vysílacího modulu (dále jen program TX) navíc konfiguruje přerušení INT0 a Wakeup funkci na pinu P03 pro probuzení z úsporného režimu vysílacího modulu. PWM modul a AD převodník jsou zapnuty a nastaveny na 8 bit režim. Časovač T2 se používá pro stanovení vzorkovací frekvence 4,8 kHz. Časovač T1 je nastaven pro odčítání hodnoty 20ms. Bezdrátová komunikace je konfigurována na komunikační frekvenci 433,0 MHz dále pak na přenos paketu dat po 32 bajtech, čtyři bajty adresy, dva bajty CRC, a na maximální vysílací výkon tj. 10 dBm. Protože piny na portu P0 obvodu nRF9E5 mají více funkcí je potřeba tento port P0 nastavit. Piny P03 a P04, jsou nastaveny jako vstupní a ostatní jako výstupní. Pin P07 je navíc nastaven jako výstup PWM modulu. U pinu P03, vysílacího modulu, je navíc nastavena funkce Wakeup.

Program TX po konfiguraci přejde do Power Down Mode (dále jen PDM) se spotřebou 2,5 uA. Aby byla spotřeba celého modulu minimální, bylo zapotřebí před PDM vypnout nízkofrekvenční zesilovač a stabilizátor NCP600 pomocí pinu P02 a P03. Protože režim PDW nevypíná AD převodník v obvodu nRF9E5, bylo nutné ho také vypnout. Z probuzení režimu PDM je nutno použít jen pin P03, který je připojen na spínač S1 tj. spínač pro zvonění. Při sepnutí spínače se vysílací modul probudí z režimu PDW a začne vysílat informaci o spuštění vyzváněcí melodie na přijímacím modulu. Poté se čeká na první příjem paketu 32 bajtů a pomocí časovače T1 se začne odpočítávat pět minut. V tomto časovém useku se může opakovaně používat spínač S1. Pokud se do této doby nepřijme žádná informace, přechází se opět do PDM. V opačném případě se zakáže přerušení od spínače S1 a u první přijaté informace se zjistí jestli jsou přijatá data „hlas“ nebo „dveře“. Data „hlas“ jsou využity pro poloduplexní komunikaci a data „dveře“ jsou použity pro sepnutí relé tj. elektronické otevírání dveří. Jestliže jsou první přijata data „hlas“ přechází se na neustálý příjem. Neustálý příjem přijímá každých 7 ms nové hodnoty dat po 32 bajtech. Přijatá data „hlas“ jsou dále odesílána do PWM modulu, zde je rekonstruován řečový signál, který je následně zesílen a reprodukován. Po každém přijetí paketu dat je spuštěn odpočet 20ms, pokud se do této doby nepřijme další paket s daty přechází se na vysílání dat „hlasu“. Vysílání také probíhá každých 7 ms

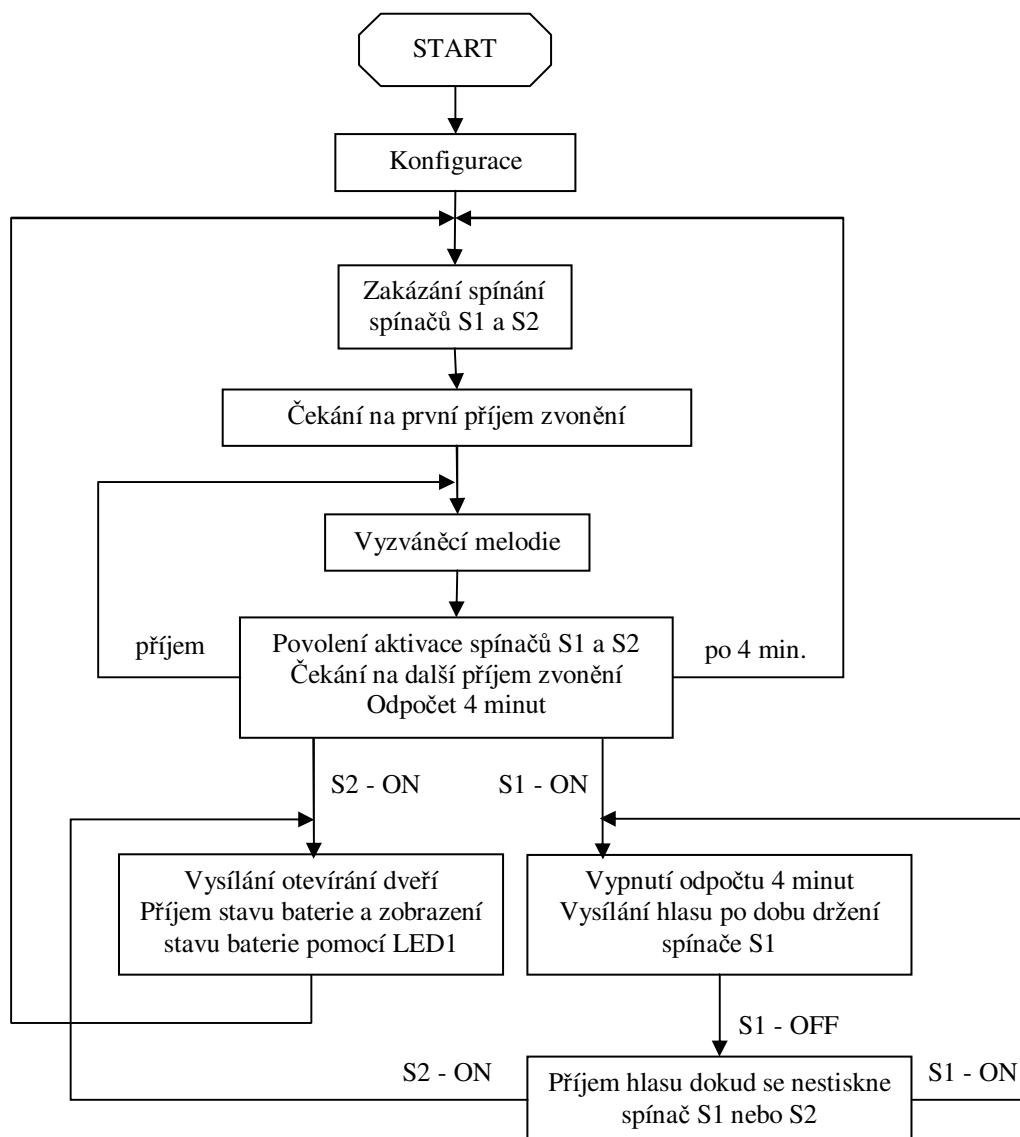
po paketech o délce 32 bajtů. Při vysílání se každou sekundu zjišťuje jestli není příjem. Je-li zjištěn příjem, vrací se program zpět na výběr „hlasu“ nebo „dveře“. Pokud není příjem zjištěn, pokračuje se ve vysílání. Kontrola příjmu při vysílání se provádí po jedné sekundě, aby nedošlo k deformaci vysílacího signálu, protože zjištění příjmu trvá 7 ms. Po tuto dobu dochází ke ztrátě jednoho paketu 32 bajtů. Při rekonstrukci řečového signálu se ztráta jednoho paketu za jednu sekundu projeví minimálně. V případě, že přijatá data nejsou „hlas“, ale „dveře“, sepne se relé pro otvírání dveří. Pomocí VDD/3 převodníku se změří napájecí napětí, vyčká se pět sekund a vypne se relé. Z naměřené hodnoty z VDD/3 převodníku se vyhodnotí, jestli není potřeba vyměnit baterie. Vyhodnocení se vyšle do přijímacího modulu. Po zjištění stavu baterie se program vrací zpět na začátek programu tj. povolení přerušení od S1 a režim PDW. Funkce programu TX je znázorněna ve vývojovém diagramu (viz Obr. 13. Vývojový diagram programu TX).



Obr. 13. Vývojový diagram programu TX

Program přijímacího modulu (dále jen program RX) po konfiguraci zakáže používání spínačů S1 a S2, tím se přejde na příjem. Zde se čeká než se přijme informace o tom, že na vysílacím modulu byl sepnut spínač S1 pro zvonění. Po příjmu „zvonění“ se spustí melodické vyzvánění, které trvá šest sekund. Po zaznění melodie se povolí používání spínačů S1 a S2, spustí se odpočet čtyř minut pomocí časovače T1 a čeká se na příjem další informace o zvonění. Po uplynutí čtyř minut, aniž by byl stisknut spínač S1 nebo S2, se zakáže používání spínačů a opět se čeká na příjem „zvonění“. Pokud se do čtyř minut přijme další informace o zvonění spustí se opět vyzváněcí melodie. Jestliže se do čtyř minut stiskne spínač S2, vyšle se

informace o sepnutí relé na vysílacím modulu, tj. otevření dveří. Přijme se stav baterie na vysílacím modulu. Stav baterie se na přijímacím modulu zobrazí pomocí LED diody. Pokud se dioda rozsvítí, je nezbytné baterie na vysílacím modulu vyměnit. Nakonec se program přesune na začátek. Pokud se stiskne spínač S1, vypne se odpočet čtyř minut, začnou se vysílat data „hlas“. Data „hlas“ se vysílají po dobu sepnutého spínače S1. Po uvolnění spínače S1 se přejde na příjem dat „hlasu“. Při příjmu se kontroluje stav obou spínačů. Je-li sepnut spínač S1 začnou se opět vysílat data „hlas“. Spínačem S1 se dá opakovaně poloduplexně komunikovat mezi přijímacím a vysílacím modulem. Ovšem je-li sepnut spínač S2, vyše se informace o sepnutí relé, přijme se stav baterie na vysílacím modulu, zobrazí se stav baterie pomocí LED diody a začne se program vykonávat od začátku. Funkce programu RX je znázorněna ve vývojovém diagramu (viz Obr. 14. Vývojový diagram programu TX ).



Obr. 14. Vývojový diagram programu TX

## 5.2 Princip vysílání a příjmu obvodu nRF9E5

Firma Nordic uvádí, že obvod nRF9E5 má maximální přenosovou rychlost 50 kbps. Ale skutečná užitečná přenosová rychlost je menší. Užitečná přenosová rychlost závisí na nastavení vysílacího a přijímacího modulu v obvodu nRF9E5. Kromě užitečné informace se přenáší kanálem preamble, adresa dat a kontrolní bity CRC.

Vysílání u obvodu nRF9E5 neprobíhá spojitě, ale po částech. Užitečná data se nahrají do modulu vysílače, k nim se přidá preamble, adresa a kontrolní bity CRC. Tento paket dat se poté začne vysílat rychlostí 100 kbps. Při vysílání se opět začne připravovat další paket. Příprava paketu je časově náročná, proto výrobce udává maximální kontinuální přenosovou rychlost 50 kbps.

Adresa dat, kontrolní bity CRC a množství užitečných dat v paketu se dá softwarově nastavit. Užitečná data jsou nastavena na maximální délku tj. 32 bajtu. Dále je přidán preamble o délce 10 bitů, adresa o délce čtyř bajtů a kontrolní bity o délce dvou bajtů. Tedy celkový vysílaný paket má délku 314 bitů. Velikost adresy a kontrolních bajtů se dá zmenšit na úkor rušení. Pokud se adresa změní, na jednobajtovou, je rušení tak velké, že se přenesou přibližně jen 20% dat. Proto je adresa dat nastavena na maximální možnou velikost. Při tomto nastavení, se vzorkovací frekvencí 4,8 kHz, je přenosová rychlost 47,1 kbps.

Nevýhodou obvodu nRF9E5 je, že po sběrnici SPI komunikuje s procesorem jak AD převodník tak vysílací modul. Při vzorkování s frekvencí 4,8 kHz se ze vstupního signálu odebere vzorek každých 208 us. Po odebrání 32. vzorku se všechny vzorky nahrají po sběrnici SPI do vysílače. Přesun vzorku do vysílače musí být dokončen před začátkem dalšího vzorkování tj. do 208 us. Pokud by se tak nestalo, AD převodník by svou komunikací po sběrnici SPI přerušil přenos 32 vzorků do vysílače. Rychlost přenosu po sběrnici SPI závisí na velikosti hodinového kmitočtu procesoru. Tento kmitočet může být maximálně 20 MHz. Sběrnice SPI komunikuje poloviční rychlostí hodinového kmitočtu tj. 10 MHz. Jeden bajt je přenesen za 800 ns. K přenosu 32 bajtů by tak stačilo 25,6 us, ale po přenosu každého jednoho bajtu je nutné se SPI sběrnice zeptat, zda byl bajt správně přenesen a zda se může poslat další bajt. Tyto instrukce dotazu na sběrnici zabírají značnou časovou prodlevu, která výrazně ovlivňuje rychlost přenosu celého paketu 32 bajtových dat po sběrnici SPI. Pokud by tato omezení nebyla, mohla by se vzorkovací frekvence zvýšit až na 5,1 kHz, kde je přenosová rychlost 50 kbps.

## 6. Závěr

Cílem mé diplomové práce bylo navrhnout a sestavit bezdrátový zvonek s digitálním přenosem hlasu. K řízení bezdrátového zvonku jsem použil čip nRF9E5 od firmy Nordic. Výhodou tohoto čipu je, že obsahuje analogově digitální převodník, pulzně šířkovou modulaci, rádiové rozhraní a mikroprocesor. K tomuto čipu byly připojeny pomocné obvody, k zajištění správné funkce bezdrátového zvonku.

Rádiové rozhraní v tomto čipu umí komunikovat v danou chvíli pouze jedním směrem. Proto je přenos hlasu proveden poloduplexně. Aby byla komunikace pro obsluhu co nejjednodušší, byl zde použit pouze jeden spínač pro poloduplexní komunikaci. Obsluha u přijímacího modulu stanoví směr a délku komunikace.

Navrhnuty byly dva moduly. Vysílací modul, který bude umístěn u dveří či brány domu a přijímací modul, který se umístí dovnitř domu. U vysílacího modulu se obsluhuje pouze jeden spínač. Tímto spínačem se spouští vyzváněcí melodie v domě. U přijímacího modulu se obsluhují dva spínače. Jeden je pro hlasovou komunikaci a druhý pro otevírání dveří či brány.

Při návrhu byl kladen důraz na co nejnižší spotřebu vysílacího modulu, který je napájen ze tří 1,5 V monočlánků. Při nečinnosti modulu je odběr pouhých 2,5 uA.

Nevýhodou tohoto bezdrátového zvonku, s čipem nRF9E5, je nízká vzorkovací frekvence 4,8 kHz. Při této frekvenci je vstupní hlasový signál omezen filtrem dolní propustí s mezním kmitočtem 2,4 kHz. Kmitočet 2,4 kHz je nedostačující pro kvalitní přenos hlasu. Nejdůležitější složky, zajišťující srozumitelnost, leží v oblasti 1 kHz až 3 kHz. Pro kvalitní přenos hlasu by bylo zapotřebí zajistit vstupní hlasový signál až od 3,4 kHz. To mi ale nedovolila přenosová rychlost čipu nRF9E5.

Výsledkem práce je plně funkční bezdrátový zvonek s poloduplexní interkomem a bezdrátovým otevíráním dveří. Deska plošného spoje byla navržena jako univerzální. Plošný spoj se dá použít pro vysílací i přijímací modul. K přijímacímu modulu je navrhnout síťový zdroj napětí. Programy do obvodů nRF9E5 byly programovány v jazyce C, ve vývojovém prostředí KEIL  $\mu$ Vision 2.

## Literatura

- [1] FRONC, V.; KLÚČIK, J. *Mikrokontroléry ATMEL s jádrem 8051*. BEN – technická literatura, Praha, 2002
- [2] SKALICKÝ, P. *Mikroprocesory řady 8051*. BEN – technická literatura, Praha, 1998
- [3] VÁCLAVÍK, R. Bezdrátový modul pro pásma 433MHz a 868MHz, [online]. Dostupnost z WWW: <<http://www.sysala.cz/Elektro/Nordic/>>
- [4] MANN, B. *C pro mikrokontroléry*. BEN – technická literatura, Praha 2003
- [5] NORDIC SEMICONDUCTOR: nRF9E5 Multiband Transceiver/MCU/ADC, Data Sheet, [online]. Dostupný z WWW: <<http://www.nvlsi.no/index.cfm?obj=product&act=display&pro=82#>>
- [6] PUNČOCHÁŘ, J. *Operační zesilovač v elektronice*. BEN – technická literatura, Praha, 2002
- [7] MATOUŠEK, D. *C pro mikrokontroléry ATMEL AT89S52*. BEN – technická literatura, Praha 2007
- [8] MANN, B. *C pro mikrokontroléry*. BEN – technická literatura, Praha 2005
- [9] ON SEMICONDUCTOR: NCP2890 1W Audio Power Amplifier, Data Sheet, [online]. Dostupný z WWW: <<http://www.onsemi.com/pub/Collateral/NCP2890-D.PDF>>
- [10] ON SEMICONDUCTOR: NCP600 High Performance Low-Power LDO Regulator with Enable, Data Sheet, [online]. Dostupný z WWW: <<http://www.onsemi.com/pub/Collateral/NCP600-D.PDF>>



## Seznam použitých zkratek

AD	Analog Digital converter
CISC	Complex Instruction Set Computer
CRC	Cyclic Redundancy Check
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESR	Equivalent Series Resistance
GFSK	Gaussian Frequency-Shift Keying
ISM	Industrial, Scientific and Medical
LED	Light-emitting diode
PWM	Pulse Width Modulation
QFN	Quad Flat No leads
RAM	Random Access Memory
RF	Radio Frequency
SMD	Surface Mount Device
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

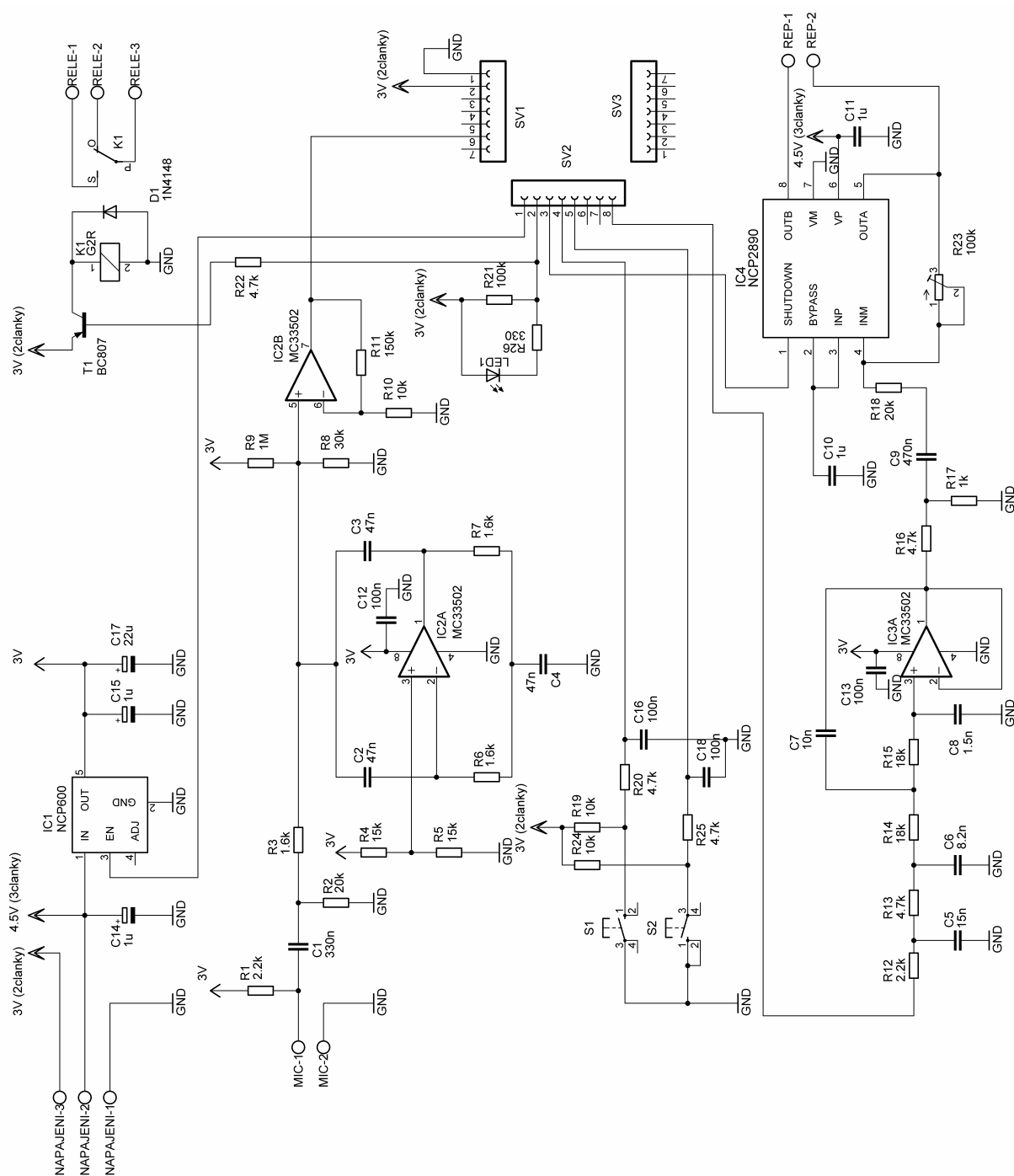
# Příloha

## I. Obsah přiloženého CD

Podklady, obsah a výsledky této bakalářské práce jsou k dispozici i v elektronické podobě na přiloženém CD. Adresářová struktura a obsah CD je následující:

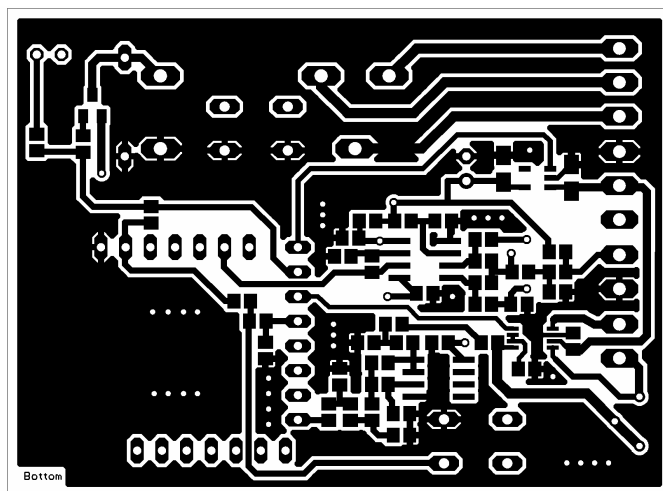
<b>Adresář</b>	<b>Obsah</b>
Diplomová práce	Tato diplomová práce
Foto	Dokumentační fotografie
METADAT	Soubor metadat
Program RX	Program pro přijímací
Program TX	Program pro vysílací modul
Seznam součástek	Seznam použitých součástek
Schémata	Schémata bezdrátového zvonku
Spoje	Výkresy plošných spojů a osazovací plán

## II. Univerzální schéma vysílacího i přijímacího modulu

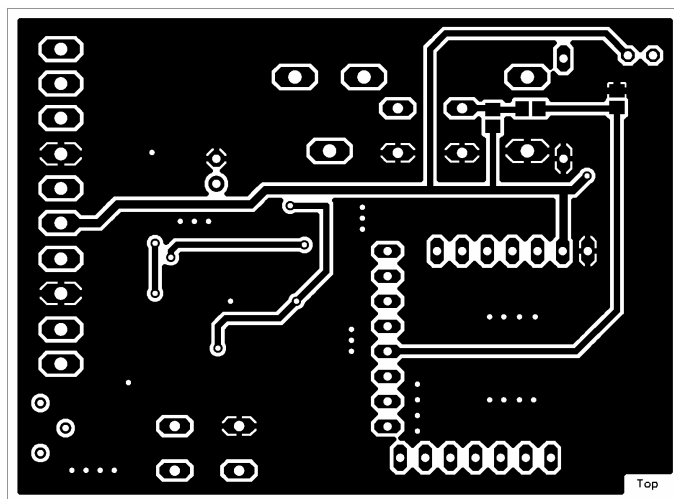


Obr. II.1 Schéma vysílacího i přijímacího modulu

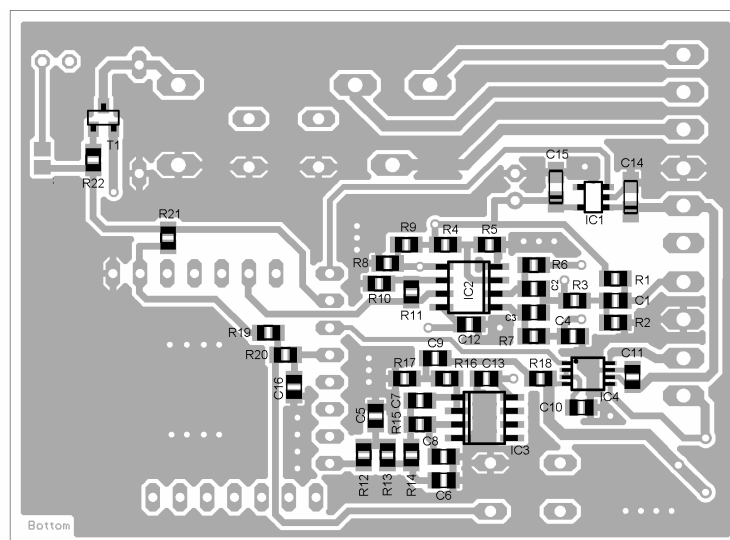
### III. Výkresy desek plošných spojů a osazovací plánek



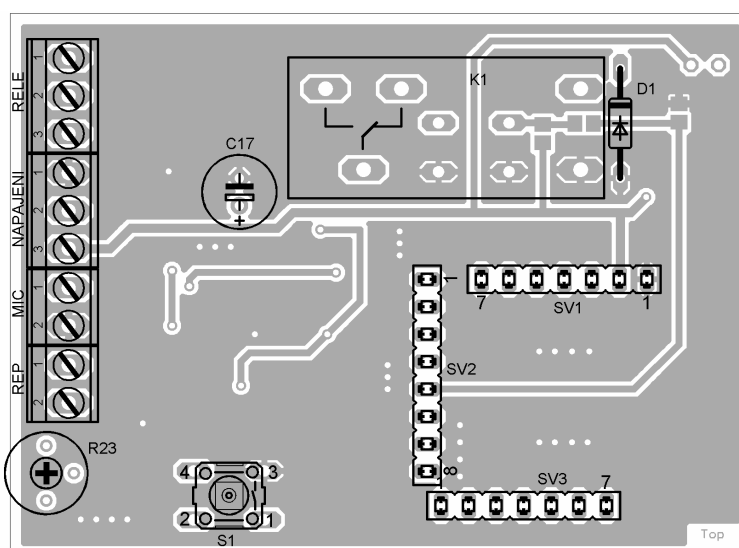
Obr III.1 Výkres univerzální desky plošných spojů modulů – strana Bottom



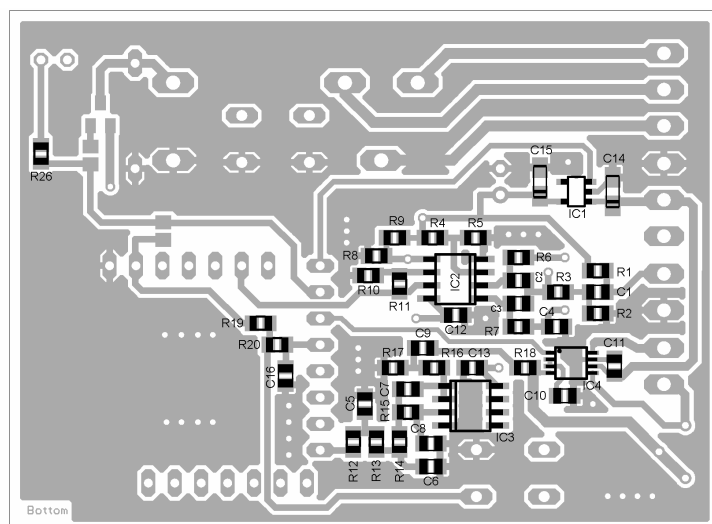
Obr III.2 Výkres univerzální desky plošných spojů modulů – strana Top



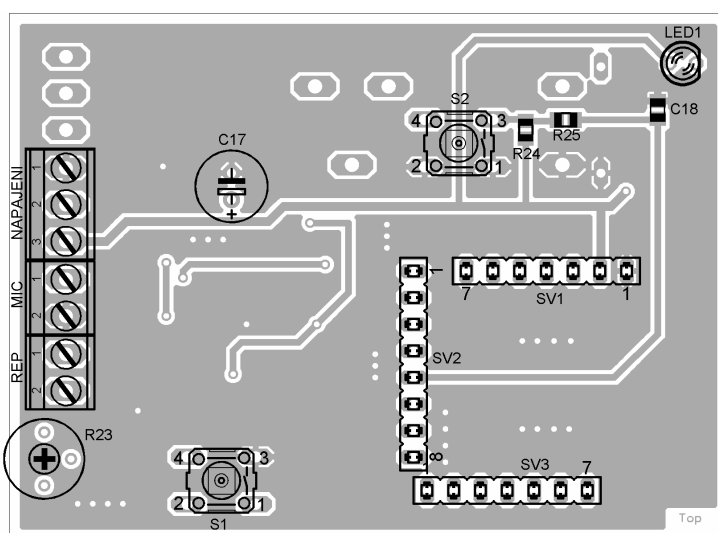
Obr. III. 3 Osazovací plánec vysílacího modulu – strana Bottom



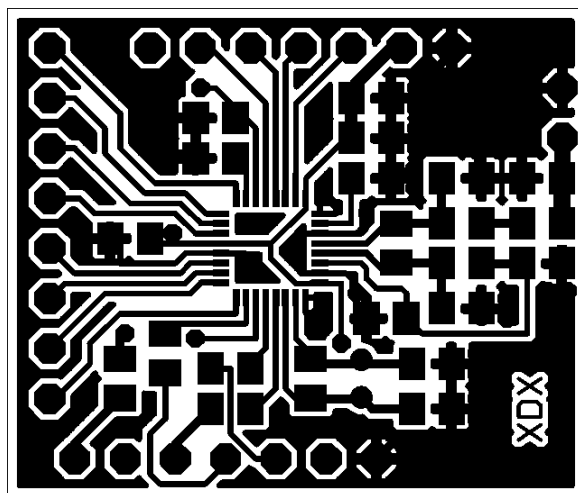
Obr. III. 4 Osazovací plánec vysílacího modulu – strana Top



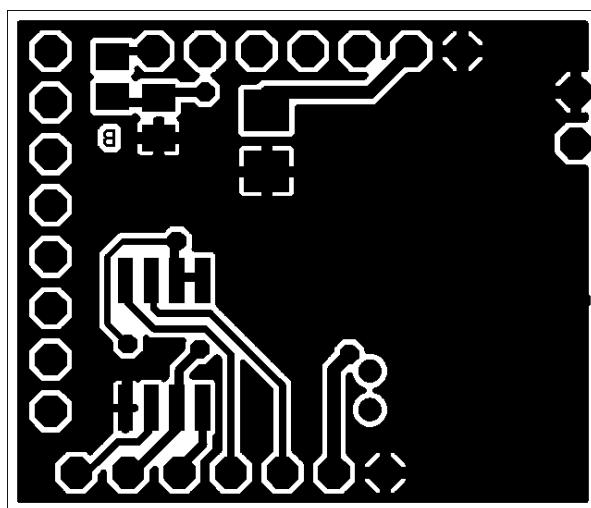
Obr. III. 5 Osazovací plánec přijímacího modulu – strana Bottom



Obr. III. 6 Osazovací plánec přijímacího modulu – strana Top



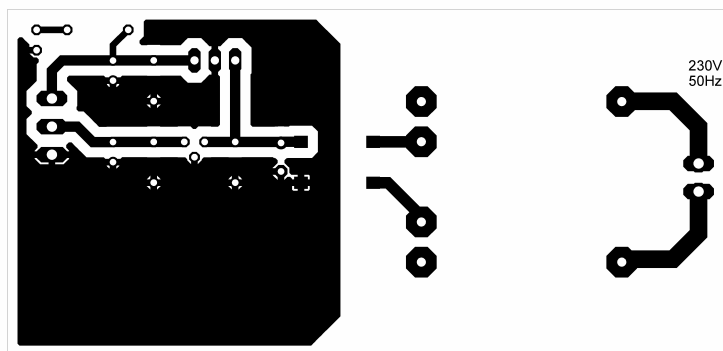
Obr. III. 7 Výkres plošných spojů řídicí jednotky nRF9E5 – strana Bottom



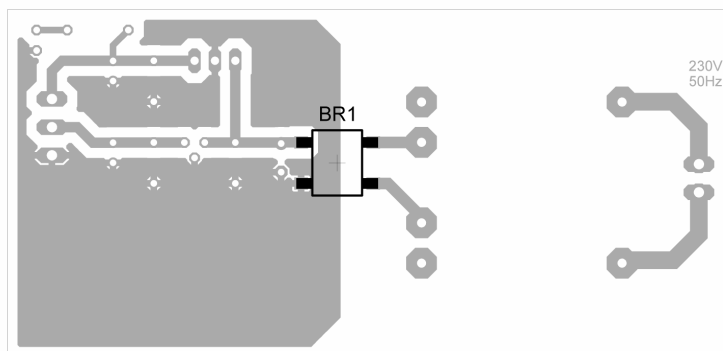
Obr. III. 8 Výkres plošných spojů řídicí jednotky nRF9E5 – strana Top



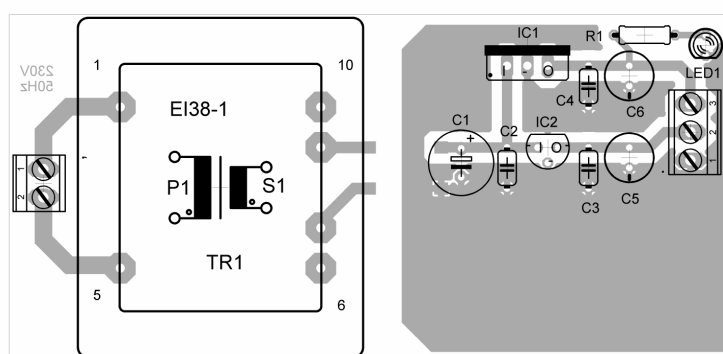




Obr. III. 11 Výkres plošného spoje síťového zdroje



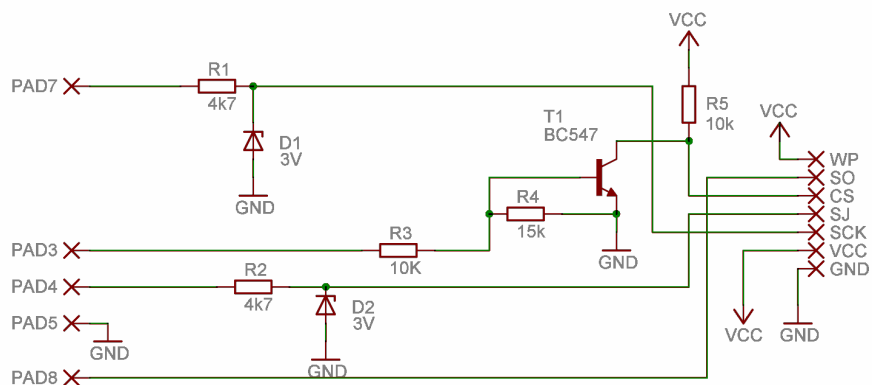
Obr. III. 12 Osazovací plánec síťového zdroje – strana Bottom



Obr. III. 13 Osazovací plánec síťového zdroje – strana Top

## IV. Programátor EEPROM paměti

Programátor pracuje přes sériovou sběrnici RS232. Vývody PADx se připojí na konektor CANON 9. Čísla vývodů jsou totožná s čísly konektoru CANON 9.



Obr.III.1 Programátor pamětí EEPROM 25AA320

## V. Výpis programu přijímacího a vysílacího modulu

Zdrojový text programu vysílacího modulu:

```
#include <reg9e5.h>

#define BUFFER 32 //Zadání velikosti zásobníku
#define BUFMASK BUFFER-1
#define INTER -33333 //Dělicí poměr pro časovač T1
#define LOW(x) (x)&0xFF //Získání dolního a horního bajtu 16bitového citace
#define HIGH(x) (x)>>8

volatile unsigned char data RxBUF[BUFFER]; //Zásobník pro přijímač
volatile unsigned char data TxBUF[BUFFER]; //Zásobník pro vysílač
volatile unsigned char RxWP, RxRP; //Ukazatelé
volatile unsigned char TxWP, TxPB;
volatile unsigned char Spat, TXorRX, VDD;
unsigned int pocitadlo, sekunda;
bit S1, ADorPWM, Odpocet, Prepnuti, Baterie;

/***** Čekací smyčka 50us *****/
void Delay50us(volatile unsigned char n)
{
    unsigned char i;
    while(n--)
        for(i=0;i<7;i++)
            ;
}

/***** Komunikace po SPI *****/
unsigned char SpiRW(unsigned char b)
{
    EXIF &= ~0x20; //Vymazání SPI přerušení
    SPI_DATA = b; //Přenos dat po SPI sběrnici
    while((EXIF & 0x20) == 0x00) //Čekání na konec přenosu
        ;
    return SPI_DATA;
}

/***** Inicializace *****/
void Init()
{
    TxPB=TxWP = 0; //Vynulování ukazatelů
    S1=0;
    Spat = 0;
    TXorRX = 0;
    Baterie = 0;

    //SPI
    SPICLK = 0x00; //Maximální rychlost SPI sběrnice
    SPI_CTRL = 0x02; //SPI připojen na nRF905 a AD ne na P0

    //TIMER2
    TR2 = 0; //Stop TIMER2
    CKCON |= 0x20; //T2M = 1 (/4 timer clock)
    RCAP2L = 0xEE; //fvz = 4,8kHz...
    RCAP2H = 0xFB; //...(65536-8e6/(4*4,8e3))=FBEE
    C_T2 = 0; //časovač
```

```

CP_RL2 = 0; //auto-reload
TF2 = 0; //příznak přetečení
TR2 = 0; //Zastavení časovače2
ET2 = 1; //Povolení přerušení časovače T2

//TIMER1
TMOD = 0x10; //Časovač, 16bitový
TL1 = LOW(INTR); //Naplnění spodních 16bit
TH1 = HIGH(INTR); //Naplnění horních 16bit
ET1 = 1;

//PWM
P0_ALT = 0x80; //Zapnutí PWM modulu
PWMCON = 0xC0; //8bit PWM
PWMDUTY = 0xFF;

// ADC
RACSN = 0;
SpiRW(WAC); //Zápis ADC konfigurace
SpiRW(0x05); //AIN0, PWR_UP = 1, VFSEL=0
SpiRW(0x09); //8bit, Right justified
RACSN = 1;

//RADIO
RACSN = 0;
SpiRW(WRC | 0x00); //Zápis konfigurace od 1B
SpiRW(0x7B); //Frekvence 433,0MHz
SpiRW(0x0C); // +10dBm, Auto_Retrans,
SpiRW(0x44); //4B pro adresu RX/TX
RACSN = 1;

RACSN = 0;
SpiRW(WRC | 0x09); //Zápis konfigurace od 10B
SpiRW(0x64); //Fosc = 20MHz, CRC = 16bit
RACSN = 1;

//Přerušení INT0
IT0 = 1; //INT0 je citlivý na nulu
EX0 = 1; //Povolení přerušení od INT0

//Přerušení AM
EX5 = 0; //Zakázání přerušení od AM

//Nastavení portu
P0_DIR = 0x08; //P03 a P04 input ostatní output
P0_ALT = 0x88; //PWM, INT0, INT1

//Port P03 - Wakeup Function z pinu P03 (Spínač S1)
REGX_MSB = 0xA0;
REGX_LSB = 0x00;
REGX_CTRL = 0x0C;
while(REGX_CTRL & 0x10)
;

EA = 1; //Povolení všech přerušení
}

/***** Nastavení vysílání baterie *****/
void InitTxBaterie()

```

```

{
    RACSN = 0;
    SpiRW(WRC | 0x03);           //Zápis konfigurace od 1B
    SpiRW(0x01);                 //1B pro přenos RX
    SpiRW(0x01);                 //1B pro přenos TX
    SpiRW(0xB1);                 //4B adresa pro RX
    SpiRW(0xB2);
    SpiRW(0xB3);
    SpiRW(0xB4);
    RACSN = 1;

    RACSN = 0;
    SpiRW(WTA | 0x00);           //Zápis konfigurace od 1B
    SpiRW(0xB1);                 //4B adresa pro TX
    SpiRW(0xB2);
    SpiRW(0xB3);
    SpiRW(0xB4);
    RACSN = 1;

    TXEN = 1;                     //ShockBurst TX
    TRX_CE = 0;
}

/***** Nastavení vysílání zvonění *****/
void InitTxZvoneni()
{
    RACSN = 0;
    SpiRW(WRC | 0x03);           //Zápis konfigurace od 1B
    SpiRW(0x01);                 //1B pro přenos RX
    SpiRW(0x01);                 //1B pro přenos TX
    SpiRW(0xA1);                 //4B adresa pro RX
    SpiRW(0xA2);
    SpiRW(0xA3);
    SpiRW(0xA4);
    RACSN = 1;

    RACSN = 0;
    SpiRW(WTA | 0x00);           //Zápis konfigurace od 1B
    SpiRW(0xA1);                 //4B adresa pro TX
    SpiRW(0xA2);
    SpiRW(0xA3);
    SpiRW(0xA4);
    RACSN = 1;

    TRX_CE = 0;
    TXEN = 1;                     //ShockBurst TX
}

/***** Nastavení příjmu hlasu *****/
void InitRxHlas()
{
    RACSN = 0;
    SpiRW(WRC | 0x03);           //Zápis konfigurace od 4B
    SpiRW(0x20);                 //32B pro přenos RX
    SpiRW(0x20);                 //32B pro přenos TX
    SpiRW(0xF1);                 //4B adresa zvonění RX
    SpiRW(0xF2);
    SpiRW(0xF3);
    SpiRW(0xF4);
    RACSN = 1;
}

```

```

RACSN = 0;
SpiRW(WTA | 0x00);           //Zápis konfigurace od 1B
SpiRW(0xF1);                 //4B adresa zvonění TX
SpiRW(0xF2);
SpiRW(0xF3);
SpiRW(0xF4);
RACSN = 1;

TXEN = 0;                     //ShockBurst RX
P00=1;                        //Spuštění vst. a vyst. filtru
P02=1;                        //Spuštění NF zesilovače
}

/***** Nastavení příjmu baterie *****/
void InitTxHlas()
{
    TXEN = 1;                  //ShockBurst TX
    TRX_CE = 0;
    P00=1;                     //Spuštění vst. a vyst. filtru
    P02=0;                     //Vypnutí NF zesilovače
}

/***** Přenos paketu dat 32B hlasu *****/
void VysilaniHlasu ()
{
    TxPB=0x00;                 //Vynulování proměnná zásobníku
    RACSN = 0;
    EXIF &= ~0x20;              //Vymazání SPI přerušení
    SPI_DATA = WTP;             //Přenos dat po SPI sběrnici
    while((EXIF & 0x20) == 0x00) //Čekání na konec přenosu
    ;
    //Přenos 32B
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[0]);        //Přenos bajtu ze zásobníku po SPI do vysilače
    while((EXIF & 0x20) == 0x00); //Čekání na konec přenosu
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[1]);
    while((EXIF & 0x20) == 0x00);
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[2]);
    while((EXIF & 0x20) == 0x00);
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[3]);
    while((EXIF & 0x20) == 0x00);
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[4]);
    while((EXIF & 0x20) == 0x00) ;
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[5]);
    while((EXIF & 0x20) == 0x00);
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[6]);
    while((EXIF & 0x20) == 0x00);
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[7]);
    while((EXIF & 0x20) == 0x00);
    EXIF &= ~0x20;
    SPI_DATA=(TxBUF[8]);
    while((EXIF & 0x20) == 0x00);
}

```

```

EXIF &= ~0x20;
SPI_DATA=(TxBUF[9]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[10]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[11]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[12]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[13]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[14]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[15]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[16]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[17]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[18]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[19]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[20]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[21]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[22]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[23]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[24]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[25]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[26]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[27]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[28]);
while((EXIF & 0x20) == 0x00);

```

```

EXIF &= ~0x20;
SPI_DATA=(TxBUF[29]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[30]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[31]);
while((EXIF & 0x20) == 0x00);
RACSN = 1;                                     //Konec přenosu 32B

TRX_CE = 1;                                   //Zapnutí vysílání
Delay50us(1);
TRX_CE = 0;                                   //Vypnutí vysílání
}

/***** Zjištění příjmu *****/
void ZjisteniPrijmu()
{
    TXEN = 0;                                //ShockBurst RX
    TRX_CE = 1;                               //Spuštění příjmu
    EX5 = 1;                                  //Povolení přerušení AM(příjmu)

    while(TxPB!=1)
    ;
    EX5 = 0;                                  //Zakázání přerušení AM(příjmu)
    TXEN = 1;                                  //ShockBurst TX
}

/***** Vysílání hlasu *****/
void Vysilani()
{
    if(sekunda == 0)                          //Zjišťování příjmu po sekunde
    {
        sekunda=50;                          //Naplnění sekundového odpočtu
        ZjisteniPrijmu();
    }
    if(Prepnuti == 0)                          //Pokud se zjistí příjem, ukončí se ...
    {                                          //...vysílání a přechází se na příjem
        while(TxPB!=32)                      //Čekání na naplnění zásobníku pro vysílání 32B hlasu...
        ;
        VysilaniHlasu();                     //...po naplnění zásobníku vysílání
        while(DR == 0)                       //Čekání na odvysílání dat
        ;
    }
}

/***** Vysílání stavu baterie *****/
void VysilaniBaterie()
{
    if(Baterie)                               //Načtení dat do vysílače při špatném stavu baterie
    {
        RACSN = 0;
        SpiRW(WTP);                          //Instrukce - zápis dat do vysílače
        SpiRW(0xBB);                          //...zápis dat - špatná baterie
        RACSN = 1;
    }
    else                                       //Načtení dat do vysílače při dobrém stavu baterie
    {
        RACSN = 0;
    }
}

```



```

    SpiRW(WTP);                                //Instrukce - zápis dat do vysílače
    SpiRW(0xCC);                                //...zápis dat - dobra baterie
    RACSN = 1;
}

TRX_CE = 1;                                    //Zapnutí vysílání
Delay50us(1);
TRX_CE = 0;                                    //Vypnutí vysílání
}

/***** Zjištění stavu napětí *****/
void StavNapeti()
{
    //Init VDD/3
    RACSN = 0;
    SpiRW(WAC);                                //Zápis ADC konfigurace
    SpiRW(0x85);                                //VDD/3, PWR_UP = 1, VFSEL=0
    SpiRW(0x09);                                //8bit, Right justified
    RACSN = 1;

    //Start převodu
    RACSN = 0;
    SpiRW(SAV | 0x08);
    RACSN = 1;

    //Čekání na konec převodu převodníku
    while(EOC == 0)
        ;

    //Čtení hodnoty z VDD/3 převodníku
    RACSN = 0;
    SpiRW(RAD);
    VDD = SpiRW(NOP);
    RACSN = 1;

    //Porovnání napětí - pokud bude napětí menší jak 1,15V na článku tak nutnost výměny baterie
    if(VDD <= 0xA1)
        Baterie = 1;
}

/***** Porovnání příchozího paketu dat 32B *****/
void HlasOrDvere()
{
    unsigned int x, y;
    EX0 = 0;                                    //Zamazání přerušení od INT0

    //Pokud jsou data rovny sepnutí rele pokud ne data jsou hlas
    if(RxBUF[0] == 0xff && RxBUF[1] == 0x01 && RxBUF[2] == 0xff && RxBUF[3] == 0x01 &&
        RxBUF[4] == 0xff && RxBUF[5] == 0x01 && RxBUF[6] == 0xff && RxBUF[7] == 0x01 &&
        RxBUF[8] == 0xff && RxBUF[9] == 0x01 && RxBUF[10] == 0xff && RxBUF[11] == 0x01 &&
        RxBUF[12] == 0xff && RxBUF[13] == 0x01 && RxBUF[14] == 0xff && RxBUF[15] == 0x01)
    {
        P01 = 0;                                //Sepnutí relé
        TR2 = 0;                                //Vypnutí časovače T2

        StavNapeti();                            //Zjištění stavu napětí

        for(x=0; x<=333U; x++)                    //5s čekání
            for(y=0; y<=5000U; y++);
    }
}

```

```

    P01 = 1;           //Vypnutí relé
    Spat = 1;          //Úsporný režim povolen
    TXorRX = 0;        //Nevysílat ani nepřijímat
}
}

/***** Příjem paketu dat 32B hlasu *****/
void PrijemHlasu()
{
    if((AM == 1) && (DR == 1))    //Čekání na přijata data
    {
        EX0 = 0;                 //Zakázání přerušení od INT0
        TR1=0;                   //Vypnutí časovače T1
        TR2 = 1;                 //Spuštění časovače T2
        TXorRX = 2;              //Neustaly příjem povolen
        Spat = 0;                //Úsporný režim zakázán
        pocitadlo = 0;           //Vynulování počítadla 5 min.
        Odpocet = 1;             //Odpocet pro neustaly příjem

        RACSN = 0;
        SpiRW(RRP);              //Instrukce - Čtení dat z přijímače
        RxRP=0x00;

        while(DR)                //Čtení dat z SPI dokud DR není 0
        {
            RxBUF[RxWP++] = SpiRW(0);    //Uložení do zásobníku
            RxWP &= BUFMASK;
        }
        RACSN = 1;
        TRX_CE = 1;

        HlasOrDvere();           //Porovnání příchozího paketu dat
    }
}

/***** První příjem paketu dat 32B hlasu *****/
void PrvniPrijemHlasu()
{
    TRX_CE = 1;                 //Spuštění příjmu - neustaly provoz
    TL1 = LOW(INTER);           //Obnovení předvolby
    TH1 = HIGH(INTER);
    TR1=1;                      //Spuštění časovače T1 - 20ms
    Spat = 1;                   //Povolení úsporného režimu
    pocitadlo = 15000U;         //Nastavení 5min odpočtu
    while(pocitadlo != 0)       //5min odpočet
    {
        PrijemHlasu();
    }
}

/***** Druhy a další příjem paketu dat 32B hlasu *****/
void NeustalyPrijemHlasu()
{
    TRX_CE = 1;                 //Spuštění příjmu - neustaly provoz
    TL1 = LOW(INTER);           //Obnovení předvolby časovače T1
    TH1 = HIGH(INTER);
    TR1=1;                      //Spuštění časovače T1

    while(Odpocet != 1)         //Pokud se do 20ms nepřijmou data přechod na vysílání
    {

```

```

    PrijemHlasu();
}
}

/***** Čekací smyčka 50us *****/
void Delay40us(volatile unsigned char n)
{
    unsigned char i;
    while(n--)
        for(i=0;i<6;i++)
            ;
}

/***** Interrupt adresy přijímače *****/
void AMRadioAdres() interrupt 11
{
    EXIF &= ~0x80;                //Vymazání AM přerušení
    TR1=0;
    Prepnuti = 1;
    TXorRX = 2;                    //Neustaly příjem povolen
}

/***** Interrupt časovače T1 *****/
void Timer1() interrupt 3
{
    TL1 = LOW(INTER);              //Obnovení předvolby
    TH1 = HIGH(INTER);

    TXorRX = 1;                    //Vysílání povoleno
    Odpocet = 1;                   //Ukončení odpočtu 20ms
    --pocitadlo;                   //5min odpočet
    --sekunda;                     //sekundový odpočet
}

/***** Interrupt časovače T2 *****/
void Timer2() interrupt 5 using 1
{
    unsigned char ADC;

    TF2 = 0;                       //Smazání přerušení časovač T2
    if(ADorPWM == 1)               //Vyber AD převodníku nebo PWM modulu
    {
        //Čtení předchozího stavu převodu AD převodníku
        RACSN = 0;
        EXIF &= ~0x20;              //Vymazání SPI přerušení
        SPI_DATA = (RAD);           //Instrukce pro přenos dat po SPI sběrnici z ADC
        while((EXIF & 0x20) == 0x00) //Čekání na konec přenosu
            ;
        EXIF &= ~0x20;              //Vymazání SPI přerušení
        SPI_DATA = (NOP);           //Instrukce pro přenos dat po SPI sběrnici z ADC
        while((EXIF & 0x20) == 0x00) //Čekání na konec přenosu
            ;
        ADC=SPI_DATA;               //Uložení hodnoty z SPI
        RACSN = 1;

        //Uložení hodnoty z ADC do zásobníku
        TxBUF[TxWP]= ADC;
        TxWP++;
        TxWP &= BUFMASK;
        TxPB++;
    }
}

```

```

//Start ADC převodu
RACSN = 0;
EXIF &= ~0x20;
SPI_DATA = (0xC0);
while((EXIF & 0x20) == 0x00)
;
RACSN = 1;
}
else
{
//Zapis do PWM
PWMDUTY = RxBUF[RxRP];
RxRP++;
RxRP &= BUFMASK;
}
}

/***** Interrupt externího INTO *****/
void Zvonení() interrupt 0
{
TR1=0;
InitTxZvonení();

RACSN = 0;
SpiRW(WTP);
SpiRW(0xAA);
RACSN = 1;

TRX_CE = 1;
Delay40us(1);
TRX_CE = 0;

while(DR == 0)
;

S1=1;
Spat = 0;
pocitadlo = 0;
TXorRX = 0;
}

/***** Hlavní program *****/
void main()
{
unsigned int x, y;

Init();
Spat = 1;

while(1)
{
if(S1)
{
S1=0;
InitRxHlas();
ADorPWM = 0;
PrvniPrijemHlasu();
}
}
}

```

if(Spat == 1)	//Úsporný režim
{	
TR1 = 0;	//Zastavení časovače T1
P00=0;	//Vypnutí vst. a vyst. filtru
P02=0;	//Vypnutí NF zesilovače
InitTxBaterie();	//Nastavení vysílání baterie
VysilaniBaterie();	//Vysílání stavu baterie
while(DR == 0)	//Čekání na odvysílání dat
;	
IE0 = 0;	
EX0 = 1;	//Povolení přerušení S1 (zvonění)
RACSN = 0;	//Vypnutí AD převodníku skrz spotřebu
SpiRW(WAC);	//Instrukce - Zápis ADC konfigurace
SpiRW(0x01);	//PWR_UP = 0
RACSN = 1;	
CK_CTRL = 0x04;	//Power Down Mode - Standby mode 2,5uA
for(x=0;x<=65U;x++)	//1s čekání na zotavení ze PDW
for(y=0;y<=5000U;y++);	
RACSN = 0;	//Spuštění AD převodníku
SpiRW(WAC);	//Zápis ADC konfigurace
SpiRW(0x05);	//AIN0, PWR_UP = 1, VFSEL=0
SpiRW(0x09);	//8bit, Right justified
RACSN = 1;	
pocitadlo=15000U;	//Naplnění 5 min. odpočtu
Spat = 0;	//Zakázání úsporného režimu
Baterie = 0;	//Vynulování stavu baterie
TXorRX = 0;	//Zakázání příjmu a vysílání
}	
if(TXorRX == 2)	//Neustaly příjem hlasu
{	
InitRxHlas();	//Nastavení příjmu hlasu
TL1 = LOW(INTER);	//Obnovení předvolby časovače T1
TH1 = HIGH(INTER);	
Odpocet=0;	
Prepnuti = 0;	
ADorPWM = 0;	//Zapnutí PWM modulu
NeustalyPrijemHlasu();	//Příjem
sekunda=50;	//Naplnění sekundového odpočtu
}	
if(TXorRX == 1)	//Neustale vysílání
{	
InitTxHlas();	//Nastavení vysílání hlasu
ADorPWM = 1;	//Zapnutí AD převodníku
Vysilani();	//Vysílání
}	
}	
}	

## Zdrojový text programu přijímacího modulu:

```
#include <reg9e5.h>

#define BUFFER 32 //Zadáni velikosti zásobníku
#define BUFMASK BUFFER-1
#define INTER -33333 //Dělnici poměr pro časovač T1
#define LOW(x) (x)&0xFF //Získání dolního a horního bajtu 16bitového citace
#define HIGH(x) (x)>>8

volatile unsigned char data RxBUF[BUFFER]; //Zásobník pro přijímač
volatile unsigned char data TxBUF[BUFFER]; //Zásobník pro vysílač
volatile unsigned char RxWP, RxRP; //Ukazatele
volatile unsigned char TxWP, TxPB;
bit ADorPWM, PrvniZvonení; //Bitové proměnné
unsigned int pocitadlo;

/***** Čekací smyčka 50us *****/
void Delay50us(volatile unsigned char n)
{
    unsigned char i;
    while(n--)
        for(i=0;i<7;i++)
            ;
}

/***** Komunikace po SPI *****/
unsigned char SpiRW(unsigned char b)
{
    EXIF &= ~0x20; //Vymazání SPI přerušení
    SPI_DATA = b; //Přenos dat po SPI sběrnici
    while((EXIF & 0x20) == 0x00) //Čekání na konec přenosu
        ;
    return SPI_DATA;
}

/***** Inicializace *****/
void Init()
{
    RxWP=RxRP = 0; //Vynulování ukazatelů
    TxWP=TxPB = 0;
    PrvniZvonení = 0;
    pocitadlo = 0;

    //SPI
    SPICLK = 0x00; //Maximální rychlost SPI sběrnice
    SPI_CTRL = 0x02; //SPI připojen na nRF905 a AD ne na P0

    //PWM
    P0_ALT = 0x80; //Zapnutí PWM modulu
    PWMCON = 0xC0; //8bit PWM
    PWMDUTY = 0xFF;

    //TIMER2
    TR2 = 0; //Stop TIMER2
    CKCON |= 0x20; //T2M = 1 (/4 timer clock)
    RCAP2L = 0xEE; //fvz = 4,8kHz...
    RCAP2H = 0xFB; //...(65536-8e6/(4*4,8e3))=FBEE
}
```

```

C_T2 = 0; //časovač
CP_RL2 = 0; //auto-reload
TF2 = 0; //příznak přetečení
ET2 = 1; //povolení přerušení časovače T2

//TIMER1
TMOD = 0x10; //Časovač, 16bitový
TL1 = LOW(INTER); //Naplnění spodních 16bit
TH1 = HIGH(INTER); //Naplnění horních 16bit
ET1 = 1;

//ADC
RACSN = 0;
SpiRW(WAC); //Zápis ADC konfigurace
SpiRW(0x05); //AIN0, PWR_UP = 1, VFSEL=0
SpiRW(0x09); //8bit, Right justified
RACSN = 1;

//RADIO
RACSN = 0;
SpiRW(WRC | 0x00); //Zápis konfigurace od 1B
SpiRW(0x7B); //frekvence 433,0MHz
SpiRW(0x0C); //+10dBm, Auto_Retrans,
SpiRW(0x44); //4B pro adresu RX/TX
RACSN = 1;

RACSN = 0;
SpiRW(WRC | 0x09); //Zápis konfigurace od 10B
SpiRW(0x64); //Fosc = 20MHz, CRC = 16bit
RACSN = 1;

//Nastavení portu
P0_DIR = 0x18; //P03 a P04 input ostatní piny output
P0_ALT = 0x80; //PWM

EA = 1; //povolení všech přerušení
}

/***** Nastavení vysílání hlasu *****/
void InitTxHlasu()
{
    RACSN = 0;
    SpiRW(WRC | 0x03); //Zápis konfigurace od 4B
    SpiRW(0x20); //32B pro přenos RX
    SpiRW(0x20); //32B pro přenos TX
    SpiRW(0xF1); //4B adresa pro RX
    SpiRW(0xF2);
    SpiRW(0xF3);
    SpiRW(0xF4);
    RACSN = 1;

    RACSN = 0;
    SpiRW(WTA | 0x00); //Zápis konfigurace od 1B
    SpiRW(0xF1); //4B adresa pro TX
    SpiRW(0xF2);
    SpiRW(0xF3);
    SpiRW(0xF4);
    RACSN = 1;

    TXEN = 1; //ShockBurst TX

```

```

ADorPWM = 1;           //Zapnutí AD převodníku
TR2 = 1;               //Spuštění časovače T2
P00=1;                //Spuštění vst. a vyst. filtru
P02=0;                //Vypnutí NF zesilovače
}

/***** Nastavení příjmu hlasu *****/
void InitRxHlasu()
{
    TXEN = 0;           //ShockBurst RX
    ADorPWM = 0;        //Zapnutí PWM modulu
    TR2 = 1;            //Spuštění časovače T2
    P00=1;              //Spuštění vst. a vyst. filtru
    P02=1;              //Spuštění NF zesilovače
}

/***** Nastavení příjmu baterie *****/
void InitRxBaterie()
{
    RACSN = 0;
    SpiRW(WRC | 0x03);  //Zápis konfigurace od 1B
    SpiRW(0x01);        //1B pro přenos RX
    SpiRW(0x01);        //1B pro přenos TX
    SpiRW(0xB1);        //4B adresa pro RX
    SpiRW(0xB2);
    SpiRW(0xB3);
    SpiRW(0xB4);
    RACSN = 1;

    RACSN = 0;
    SpiRW(WTA | 0x00);  //Zápis konfigurace od 1B
    SpiRW(0xB1);        //4B adresa pro TX
    SpiRW(0xB2);
    SpiRW(0xB3);
    SpiRW(0xB4);
    RACSN = 1;
    TXEN = 0;           //ShockBurst TX
}

/***** Nastavení vysílání dveře (relé) *****/
void InitTxDvere()
{
    RACSN = 0;
    SpiRW(WRC | 0x03);  //Zápis konfigurace od 4B
    SpiRW(0x20);        //32B pro přenos RX
    SpiRW(0x20);        //32B pro přenos TX
    SpiRW(0xF1);        //4B adresa zvonění RX
    SpiRW(0xF2);
    SpiRW(0xF3);
    SpiRW(0xF4);
    RACSN = 1;

    RACSN = 0;
    SpiRW(WTA | 0x00);  //Zápis konfigurace od 1B
    SpiRW(0xF1);        //4B adresa zvonění TX
    SpiRW(0xF2);
    SpiRW(0xF3);
    SpiRW(0xF4);
    RACSN = 1;
}

```



```

TXEN = 1; //ShockBurst TX
TR2 = 0; //Zastavení časovače T2
P00=0; //Vypnutí vst. a vyst. filtru
P02=0; //Vypnutí NF zesilovače
TRX_CE = 0;
}

/***** Vyzváněcí melodie trvající 6s *****/
void Zvoneni()
{
    unsigned int c;
    unsigned char d,e;

    TR2 = 1; //Spuštění časovače2
    P00=1; //Spuštění vst. a vyst. filtru
    P02=1; //Spuštění NF zesilovače

    for(e=0;e<3;e++)
    {
        for(c=0;c<20000U;c++)
        {
            for(d=0;d<4;d++)
            {
                RxBUF[RxWP++]=0xff;
                RxWP &= BUFMASK;
            }
            for(d=0;d<4;d++)
            {
                RxBUF[RxWP++]=0x00;
                RxWP &= BUFMASK;
            }
            if((P03==0) || (P04 == 0))break; //Přerušení od spínače S1 nebo S2
        }
        for(c=0;c<20000U;c++)
        {
            for(d=0;d<2;d++)
            {
                RxBUF[RxWP++]=0xff;
                RxWP &= BUFMASK;
            }
            for(d=0;d<2;d++)
            {
                RxBUF[RxWP++]=0x00;
                RxWP &= BUFMASK;
            }
            if((P03==0) || (P04 == 0))break; //Přerušení od spínače S1 nebo S2
        }
        if((P03==0) || (P04 == 0))break; //Přerušení od spínače S1 nebo S2
    }

    TR2 = 0; //Vypnutí časovače2
    P00=0; //Vypnutí vst. a vyst. filtru
    P02=0; //Vypnutí NF zesilovače
}

/***** Přenos paketu dat 32B hlasu *****/
void VysilaniHlasu()
{
    TxPB=0x00; //Vynulování proměnné zásobníku

```

```

RACSN = 0;
EXIF &= ~0x20;
SPI_DATA = WTP;
while((EXIF & 0x20) == 0x00)
;
//Prenasena data 32B
EXIF &= ~0x20;
SPI_DATA=(TxBUF[0]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[1]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[2]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[3]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[4]);
while((EXIF & 0x20) == 0x00) ;
EXIF &= ~0x20;
SPI_DATA=(TxBUF[5]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[6]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[7]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[8]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[9]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[10]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[11]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[12]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[13]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[14]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[15]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[16]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[17]);
while((EXIF & 0x20) == 0x00);

```

```

// Vymazání SPI přerušení
//Přenos dat po SPI sběrnici
//Čekání na konec přenosu

```

```

//Přenos bajtu ze zásobníku po SPI do vysílače
//Čekání na konec přenosu

```

```

EXIF &= ~0x20;
SPI_DATA=(TxBUF[18]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[19]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[20]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[21]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[22]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[23]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[24]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[25]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[26]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[27]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[28]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[29]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[30]);
while((EXIF & 0x20) == 0x00);
EXIF &= ~0x20;
SPI_DATA=(TxBUF[31]);
while((EXIF & 0x20) == 0x00);
RACSN = 1;                                     //Konec přenosu 32B

TRX_CE = 1;                                   //Zapnutí vysílání
Delay50us(1);
TRX_CE = 0;                                   //Vypnutí vysílání
}

```

/\*\*\*\*\*\* Přenos paketu dat 32B dveře \*\*\*\*\*/

```
void VysilaniDvere()
```

```

{
    unsigned char p;

    RACSN = 0;
    SpiRW(WTP);                               //Přenos dat po SPI sběrnici

    for(p=0;p<16;p++)                          //Načtení 32B do vysílače
    {
        SpiRW(0xff);
    }
}

```

```

    SpiRW(0x01);
}
RACSN = 1;
TRX_CE = 1;                                //Zapnutí vysílání
Delay50us(1);
TRX_CE = 0;                                //Vypnutí vysílání
}

/***** Příjem stavu baterie a zobrazení *****/
void PrijemBaterie()
{
    unsigned char q;

    TRX_CE = 1;                                //Spuštění příjmu - neustaly provoz
    while((AM == 0) || (DR == 0))              //Čekání na příjem
        ;
    RACSN = 0;
    SpiRW(RRP);                                //Instrukce - čtení dat z přijímače
    q = SpiRW(0);                              //Přečtení přijatého bajtu
    RACSN = 1;
    TRX_CE = 0;

    if(q == 0xBB)                              //Signalizace při špatném stavu baterie
        P01=0;
    if(q == 0xCC)                              //Signalizace při dobrém stavu baterie
        P01=1;
}

/***** Příjem paketu dat 32B hlasu *****/
void PrijemHlasu()
{
    TRX_CE = 1;                                //Spuštění příjmu - neustaly provoz
    while(DR == 0)                              //Čekání na přijata data
        ;
    RACSN = 0;
    SpiRW(RRP);                                //Čtení dat z SPI
    RxRP=0x00;
    while(DR)                                  //Čtení dat 32B z SPI dokud DR není 0
    {
        RxBUF[RxWP++]=SpiRW(0);
        RxWP &= BUFMASK;
    }
    RACSN = 1;
    TRX_CE = 1;
}

/***** Příjem 1B zvonění *****/
void PrijemZvoneni()
{
    unsigned char b;

    //Nastavení přijímači adresy
    RACSN = 0;
    SpiRW(WRC | 0x03);                        //Zápis konfigurace od 4B
    SpiRW(0x01);                              //1B pro přenos RX
    SpiRW(0x01);                              //1B pro přenos TX
    SpiRW(0xA1);                              //4B adresa zvonění
    SpiRW(0xA2);
    SpiRW(0xA3);
    SpiRW(0xA4);
}

```

```

RACSN = 1;

TXEN = 0; //ShockBurst TX
TRX_CE = 1; //Spuštění příjmu - neustaly provoz

while((AM == 0) || (DR == 0)) //Čekání na příjem
    if(((P04 == 0) || (P03 == 0)) && PrvniZvoneni)break; //Spínače S1 a S2 aktivní po ...
                                                    //.. prvním zvonění po dobu 4min

PrvniZvoneni = 1; //Nastavení signalizace prvního příjmu zvolení
TR1 = 1; //Spuštění časovače T1

RACSN = 0;
SpiRW(RRP); //Čtení dat z SPI
b = SpiRW(0); //Přečtení přijatého bajtu
RACSN = 1;
TRX_CE = 0;

if(b == 0xAA) //Kontrola přijatého bajtu
{
    TRX_CE = 0; //Vypnutí příjmu
    Zvoneni(); //Podprogram vyzváněcího tónu
}

if(P04==0) //Zjištění stavu spínače S1
{
    InitTxDvere(); //Inicializace vysílače dveře
    VysilaniDvere(); //Vyslání paketu dat dveře
    while(DR == 0) //Čekání na odvyzlání dat
        ;
    PrvniZvoneni = 0; //Vynulování signalizace prvního příjmu zvonění
    TR1 = 0; //Zastavení čítače T1
    pocitadlo = 0; //Vynulování 4 min. počítadla
    InitRxBaterie(); //Inicializace přijímače baterie
    PrijemBaterie(); //Příjem stavu baterie dveřního modulu
}
}

/***** Interrupt časovače T1 *****/
void Timer1() interrupt 3
{
    TL1 = LOW(INTER); //Obnovení předvolby
    TH1 = HIGH(INTER);

    ++pocitadlo;
    if(pocitadlo == 12000U) //Odpočet 4 min.
    {
        PrvniZvoneni=0; //Vynulování signalizace prvního příjmu zvonění po 4 min.
        TR1 = 0; //Zastavení čítače T1 po uplynutí 4min.
    }
}

/***** Interrupt časovače T2 *****/
void Timer2() interrupt 5 using 1
{
    unsigned char ADC;

    TF2 = 0; //Smazání přerušení časovač T2
    if(ADorPWM == 1) //Vyber AD převodníku nebo PWM modulu
    {
        //Čtení předchozího stavu převodu AD převodníku
    }
}

```

```

RACSN = 0;
EXIF &= ~0x20;
SPI_DATA = (RAD);
while((EXIF & 0x20) == 0x00)
;
EXIF &= ~0x20;
SPI_DATA = (NOP);
while((EXIF & 0x20) == 0x00)
;
ADC=SPI_DATA;
RACSN = 1;

//Uložení hodnoty z ADC do zásobníku
TxBUF[TxWP]= ADC;
TxWP++;
TxWP &= BUFMASK;
TxPB++;

//Start ADC převodu
RACSN = 0;
EXIF &= ~0x20;
SPI_DATA = (0xC0);
while((EXIF & 0x20) == 0x00)
;
RACSN = 1;
}
else
{
//Zapis do PWM
PWMDUTY = RxBUF[RxRP];
RxRP++;
RxRP &= BUFMASK;
}
}

/***** Hlavni program *****/
void main()
{
    unsigned int f;

    Init();
    while(1)
    {
        do
            PrijemZvoneni();
        while(P03 == 1);

        PrvniZvoneni = 0;
        TR1 = 0;
        pocitadlo = 0;

        while(P04==1)
        {
            InitTxHlasu();
            do
                if(TxPB == 32)
                    VysilaniHlasu();
            while(P03==0);
            TR2 = 0;
        }
    }
}

```

//Vymazání SPI přerušení  
//Instrukce pro přenos dat po SPI sběrnici z ADC  
//Čekání na konec přenosu  
  
//Vymazání SPI přerušení  
//Instrukce pro přenos dat po SPI sběrnici z ADC  
//Čekání na konec přenosu  
  
//Uložení hodnoty z SPI  
  
//Vymazání SPI přerušení  
//Příkaz = Start čtení z AD0  
//Čekání na konec přenosu příkazu  
  
//Načtení hodnoty ze zásobníku do PWM  
  
//Inicializace  
//Nekonečná smyčka  
  
//Příjem zvonění dokud se nestiskne spínač S1  
  
//Vynulování signalizace prvního příjmu zvonění  
//Zastavení časovače T1  
//Vynulování 4 min. počítadla  
  
//Komunikace dokola dokud se nestiskne spínač S2  
  
//Nastavení vysílání hlasu  
//Vysílání dokud se neuvolní spínač S1  
//Čekání na naplnění zásobníku pro vysílání 32B hlasu...  
//...po naplnění zásobníku vysílání  
  
//Vypnutí časovače T2

InitRxHlasu();	//Nastaveni prijmu hlasu
do	//Prijem dokud se nestiskne spínač S1 nebo S2
PrijemHlasu();	
while((P03==1) && (P04==1));	
TR2 = 0;	//Vypnutí časovače T2
}	
InitTxDvere();	//Nastaveni vysílání dveří
for(f=0;f<200;f++)	//200x vysílání
{	
VysilaniDvere();	
while(DR == 0)	//Čekání na odvyslání dat
;	
}	
InitRxBaterie();	
PrijemBaterie();	//Prijem stavu napětí modulu dveří
}	
}	